NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

VIRTUAL MACHINE MODULES FOR USE BY DOD C4I SUPPORT CENTERS

by

Luis M. Tiglao

September 2010

Thesis Co-Advisors: Albert Barreto
Karl Pfeiffer

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2010	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Virtual Machine Modules for Use by DoD C4I Support Centers		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Luis M. Tiglao			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A .			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release, distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

The mission of DoD C4I Support Centers (DCSCs) is to provide C4I application support to various communities, such as Operations and Experimentation, Training, Acquisition, and Analysis and Assessment. In order to support its respective communities, DCSCs purchases computing equipment (laptops, servers, switches) to create models and/or simulations (M&S) of current IT capabilities of the operating forces. Many times, DCSCs is required to stay up-to-date with DoD operating forces, which leads to excessive expenditures of equipment, maintenance, storage, and personnel costs.

Virtual Machines, or software implementations of real computer machines, aim to address these issues plus more. Three benefits of using virtual machine environments in M&S are: One, it reduces purchasing and maintenance costs of IT systems. Two, it provides a scalable environment that does not require excessive manpower or time to establish. Three, it drastically reduces the footprint required for established environments and gets rid of storage requirements for older systems.

This thesis focuses on the benefits and the methods needed to achieve the benefits of using commercial-off-the-shelf (COTS) virtual environments for C4I modeling and simulations. It will also introduce a modularized and reusable methodology when using the DoD Verification, Validation, and Accreditation (VV&A) process.

14. SUBJECT TERMS VM C4I Virtual Machine Virtualization Module Reuse			**15. NUMBER OF PAGES** 127
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

VIRTUAL MACHINE MODULES
FOR USE BY DOD C4I SUPPORT CENTERS

Luis M. Tiglao
Captain, United States Marine Corps
B.S., University of California in Irvine, 1998

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN INFORMATION SYSTEM TECHNOLOGY
and
MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(COMMAND, CONTROL & COMMUNICATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
September 2010

Author: Luis M. Tiglao

Approved by: Albert Barreto
Thesis Co-Advisor

Karl Pfeiffer
Thesis Co-Advisor

Dan Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The mission of DoD C4I Support Centers (DCSCs) is to provide C4I application support to various communities, such as Operations and Experimentation, Training, Acquisition, and Analysis and Assessment. In order to support its respective communities, DCSCs purchases computing equipment (laptops, servers, switches) to create models and/or simulations (M&S) of current IT capabilities of the operating forces. Many times, DCSCs is required to stay up-to-date with DoD operating forces, which leads to excessive expenditures of equipment, maintenance, storage, and personnel costs.

Virtual Machines, or software implementations of real computer machines, aim to address these issues plus more. Three benefits of using virtual machine environments in M&S are: One, it reduces purchasing and maintenance costs of IT systems. Two, it provides a scalable environment that does not require excessive manpower or time to establish. Three, it drastically reduces the footprint required for established environments and gets rid of storage requirements for older systems.

This thesis focuses on the benefits and the methods needed to achieve the benefits of using commercial-off-the-shelf (COTS) virtual environments for C4I modeling and simulations. It will also introduce a modularized and reusable methodology when using the DoD Verification, Validation, and Accreditation (VV&A) process.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

DoD	Department of Defense
DCSC	DoD C4I Support Center
C4I	Command, Control, Communication, Computers, and Intelligence
MCTSSA	Marine Corps Tactical Systems Support Activity
MCCDC	Marine Corps Combat Developmental Command
M&S	Model and Simulation
OFTSSC	Operating Forces Tactical Systems Support Center
PESG	Program & Engineering Support Group
T&CG	Test & Certification Group
TI&SG	Technical Infrastructure & Support Group
V&V	Validation and Verification
VV&A	Validation, Verification, and Accreditation

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I must first thank God for providing me the strength, will, and knowledge to complete this thesis. You are my Rock and my Deliverer.

I then want to thank my dear wife, Heidi, for her support and encouragement throughout my time at the Naval Postgraduate School. You are my best friend and I am blessed to have a wonderful wife such as you.

I also want to thank my advisors, Lt. Col. Karl Pfeiffer and Mr. Albert Barreto. Lt. Col. Pfeiffer, your insight and experience have allowed me to develop a thesis worthy of the Naval Postgraduate School. Mr. Barreto, your patience and passion for virtualization technologies spurred me to pursue this thesis.

I also wish to give special mention to Dr. Donald Brutzman and LTC Robert Shearer for providing feedback on the Modeling and Simulations portions of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

This thesis investigates the applicability of using C4I virtual machines models (i.e., software implementations of real computer systems) in DoD C4I Support Centers. It focuses on the benefits (and the methods to achieve the benefits) of using commercial-off-the-shelf virtual machines for C4I modeling and simulations. It also introduces a modularized methodology when using the DoD Verification, Validation, and Accreditation (VV&A) process.

A. DISCUSSION

The mission of DoD C4I Support Centers (DCSC) is to provide C4I application support to various communities, such as Operations and Experimentation, Training, Acquisition, and Analysis and Assessment. In order to support their respective communities, DCSCs purchase computing equipment (e.g., laptops, servers, switches) to create models and/or simulations of current IT capabilities of the operating forces. Many times, DCSCs are required to stay up to date with DoD operating forces, which lead to excessive expenditures of equipment, maintenance, storage, and personnel costs.

B. PURPOSE

Three benefits of using virtual machine environments in M&S are: One, it reduces purchasing and maintenance costs of IT systems. Two, it provides a scalable environment that does not require excessive manpower or time to establish. Three, it drastically reduces the footprint required for established environments and gets rid of facility storage requirements for older systems. This thesis proposes exploiting these benefits for DCSCs, as well as the benefits of employing reusable VM modules and reusable VV&A documentation.

C. RESEARCH METHODOLOGY

Three principal research methods were used in order to develop this thesis. They provided the base knowledge and expertise that laid the foundations of this paper.

Literature Review: Conduct a literature review of books, journals, Internet articles, and previous research.

Hands-on Experience: Volunteer time at Virtualization and Cloud Computing Lab at the Naval Postgraduate School for hands-on experience with virtual machines.

Conference Attendance: Attend Trade Conferences (i.e., VMWorld 2010) to stay current on virtualization technologies and best practices. Also initiate and coordinate USMC attendees for discussion of current and future virtualization efforts within the Marine Corps.

D. ORGANIZATION

This thesis is organized in the following chapters:

Chapter I provides the introduction and overview of this thesis.

Chapter II describes Modeling and Simulations (M&S) and the associated Verification, Validation, and Accreditation (VV&A) process.

Chapter III describes the various types of Virtual Machines along with their strengths and weaknesses.

Chapter IV describes commercial-off-the-shelf (COTS) systems and related DoD policies and learned lessons.

Chapter V combines the concepts in the previous three chapters, analyzes them, and presents recommended practices for DoD C4I Support Centers by using COTS VMs as C4I Models.

Chapter VI is a use case that implements concepts found in the previous chapter on an existing DoD C4I Support Center.

Chapter VII serves to conclude this thesis as well as give recommendations for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. OVERVIEW: MODELS AND SIMULATIONS

Models are “physical, mathematical, or otherwise, logical representation of a system, entity, phenomenon, or process” (DoD Directive 5000.59, 2007). Their primary uses include training and helping to make managerial, operational, or technical decisions. They are used throughout the DoD to include training facilities, operations research, acquisitions, etc. This chapter will first discuss the importance and limitations of M&S for C4I systems. It then introduces the DoD Modeling & Simulation (M&S) program, along with the associated Verification, Validation, and Accreditation (VV&A) process. It then shows the steps required to accredit an M&S. This chapter concludes by examining the risks incurred by ignoring the use of Modeling and Simulations by DoD C4I Modeling and Simulation Centers.

A. COMMON TERMINOLOGY

Accreditation – “The official certification that a model, simulation, or federation of models and simulations and its associated data are acceptable for use for a specific purpose.” (DoD Directive 5000.59, 2007)

Credibility – The amount and confidence that a user sees in the M&S in how it meets requirements. It is generated by the model’s accuracy, capability, correctness, and usability.

Model - “A physical, mathematical, or otherwise, logical representation of a system, entity, phenomenon, or process.” (DoD Directive 5000.59, 2007)

Simulation – "A method for implementing a model over time. Also, a technique for testing, analysis, or training in which real-world systems are used, or where real-world and conceptual systems are reproduced by a model." (DoD Directive 5000.59, 2007)

Validation – "The process of determining the degree to which a model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model." (DoD Directive 5000.59, 2007)

Verification – "The process of determining that a model implementation and its associated data accurately represents the developer's conceptual description and specifications." (DoD Directive 5000.59, 2007)

B. WHY USE MODELING?

First and foremost, it is DoD policy that "Models, simulations, and associated data used to support DoD processes, products, and decisions shall undergo verification and validation (V&V) throughout their lifecycles" (DoD Directive 5000.61, 2009). The intent and primary goal of a model, however, is to aid in training or in making managerial, operational, or technical decisions. This can be in the form of prototypes, simulators, or stimulators. According to the M&S University (MSIAC, 2009):

> M&S provides a method for training individuals and units in a safe environment, while optimizing the expenditure of your precious, limited resources. Military analysts use M&S to help shape the size, composition, and structure of forces to meet national military requirements, and to assess the sufficiency of operational plans. The military acquisition community uses M&S: (1) to evaluate requirements for new systems

> and equipment; (2) to conduct research, development and analysis activities; (3) to develop digitized prototypes and avoid the building of costly full scale mockups; and (4) to plan for efficient production and sustainment of the new systems and equipment when employed in the field.

Models are not meant to reflect every nuance of the real world environment in which a system is meant to operate in. This is both costly and nearly impossible. In fact, models purposely leave out some variables of the real world in order to isolate and identify problems of interest in a system. <u>A good model isn't determined by how closely it mirrors a system but rather, by its ability to aid the decision-making process</u>. Models accomplish this goal by creating a repeatable, inexpensive, controlled, safe, and rapid environment that is similar to the environment in which a system will operate. These individual traits are explained further below.

Repeatable - Models provide repeatability by re-creating the important aspects of a real environment. When a simulated scenario is finished, the simulation can be rerun with minimal setup to further testing. For example, a C4I program can be installed on a modeled computing environment in order to test its security measures against a certain hacking technique. Once accomplished, the model can be recreated to test a different hacking technique.

Inexpensive - Models provide inexpensive alternatives to real system environments. This applies to both initial acquisition costs as well as total cost of ownership. Models are cheaper to create and maintain. For instance, the cost of energy, maintenance, manpower, storage, and destruction are significantly cheaper for models than real objects or systems.

Controlled – Due to the repeatability attribute of models, an operation can be duplicated multiple times in a controlled environment. For example, what if the Marine Corps wanted to know how a new C4I system interoperates with the rest of its systems? A model can be created of the operating environment to determine this. The model can have different combinations of different systems interacting with the new C4I program in a controlled environment.

Safe – The capabilities of a new system can be tested without the risk of losing actual resources. For example, the effects of a computer virus towards a C4I system can be tested in a modeled environment without the fear of infecting or affecting real systems permanently.

Rapid - Tests can be run in less than real time through automation and/or time compression. For example, a program can simulate the inputs of dozens of users to test a C4I system’s ability to deal with synchronization and throughput. Models also provide a rapid environment in the sense that models are easier and faster to develop than actual systems, since only the tested aspects of the system are being recreated.

Despite the advantages listed, models do have a major disadvantage: They are not real things. If a test engineer needed to test application compatibility with a specific computer (e.g., Dell Inspiron 1545, Dell PowerEdge T105 Server) or other specific hardware (e.g., Linksys Etherfast NIC, Netgear RangeMax Wireless Card), a model of a generic computer will not fulfill his requirements. Such tests can only be run on the actual hardware itself, due to firmware nuances or hardware specificities. Performing a hardware compatibility test still requires acquisition of the actual hardware itself. In such a scenario, models are not the answer.

C. MODELING AND SIMULATION PROCESS

The DoD has an established process for modeling and simulation (Figure II-1). This process creates a disciplined approach to ensure that a model (or simulation) meets the needs that it was created for. It is not unlike most system or software development processes. The VV&A Implementation Handbook provides more detail on the steps but they are summarized here for ease of reference.

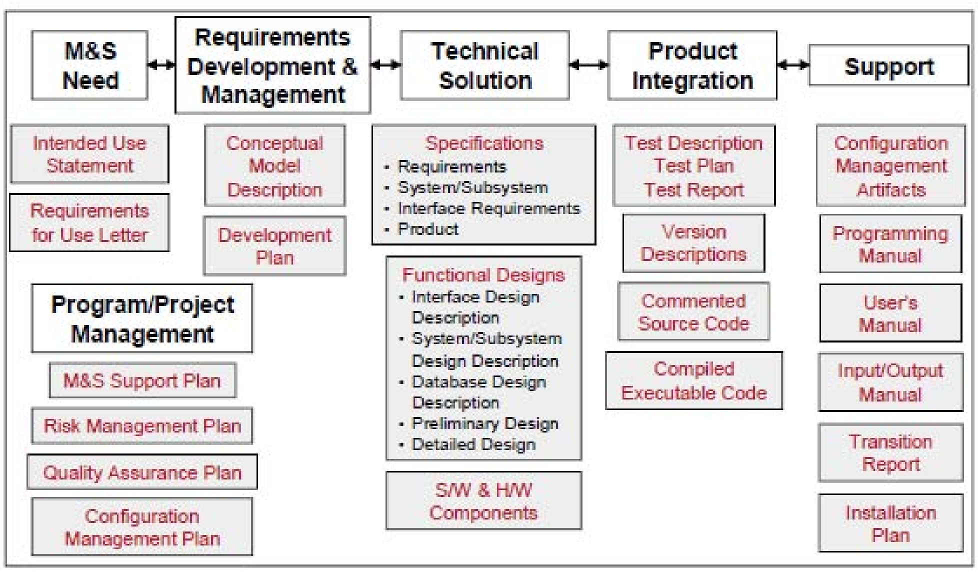

Figure II-1: M&S Process and Product Outputs (From Navy Modeling and Simulation Management Office, 2004)

1. M&S Need

As with any project, understanding the need of the M&S is crucial in developing its requirements. Why is M&S needed in the first place? This determines the intended use of the M&S and helps to define requirements.

2. Requirements Development and Management

This step determines the set of requirements a model must satisfy. Once requirements are refined and objectives determined, a model or simulation could be developed properly to meet those needs. It has two major steps: The creation of the **conceptual model** and the **development plan.** The former step helps create a mutual understanding between user and developer. It helps validate that the developer understands the intended use of the model or simulation. The latter step (i.e., development plan) determines the development method, resource allocations, schedules, etc., that will allow a model to meet the given requirements.

3. Technical Solution

This step involves **design development** and **implementation.** Design development translates the *conceptual model* into design specifications. These specifications will support the actual implementation of the model or simulation through software and/or hardware. This step also entails the actual implementation of the M&S (e.g., code development and documentation).

4. Product Integration

This step completes the integration of the different M&S components and modules. Once integrated, test scenarios will be ran and results recorded to support implementation verification and results validation of the M&S.

5. Support

As with any system, support development, deployment, and maintenance of the M&S is required. This can take the form of configuration management, training, technical support, and disposal. This process, especially configuration management, is important for accreditation. Accreditation applies to a specific version number and any changes to the accredited version must be noted.

6. Project Management

This is an overarching process that ensures that the entire M&S process progresses smoothly. It develops the M&S support, risk management, quality assurance, and configuration management plan.

D. WHAT IS THE VV&A PROCESS?

VV&A is short for Verification, Validation, and Accreditation. It is the DoD's policy that "models, simulations, and associated data used to support DoD processes, products, and decisions shall undergo verification and validation throughout their lifecycles… and shall be accredited for an intended use" (DoD Directive 5000.61, 2009). In addition to it being a DoD policy, it is a series of steps that provides credibility for a model and its simulation. Credibility is important because a user, tester, program officer, etc. needs to feel confident that tests run on a modeled system properly apply to expected tests and results for an actual system. VV&A is inextricably linked to the M&S Process (Figure II-2) and should be executed concurrently with it. The VV&A Implementation Handbook provides more detail on each of the steps, which are summarized here for ease of reference.

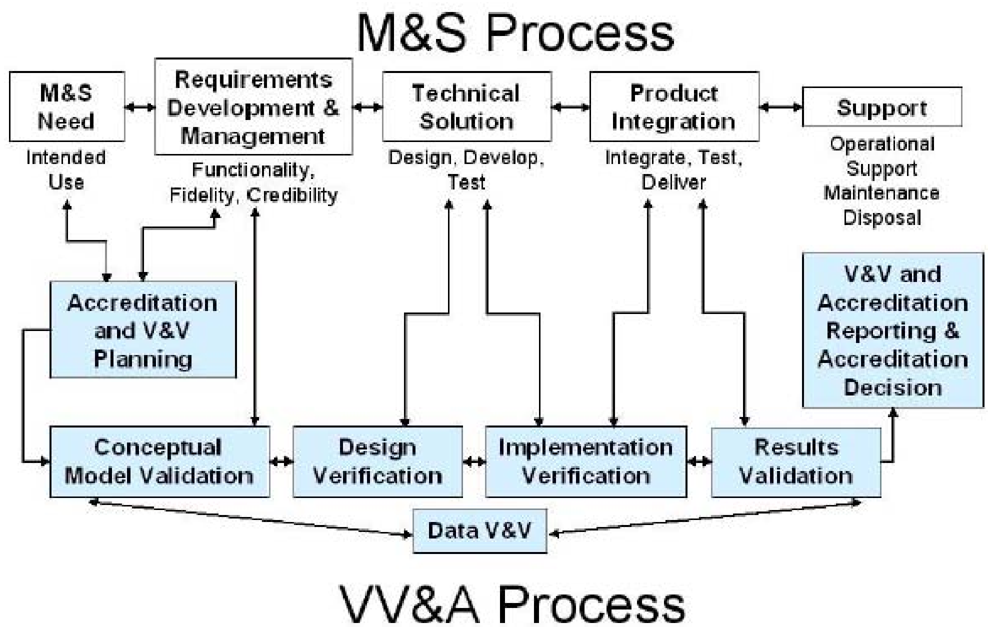

Figure II-2: M&S and VV&A Process Relationship (From Navy Modeling and Simulation Management Office, 2004)

1. Accreditation Planning

This step establishes the acceptability criteria that the M&S must meet for formal accreditation. It consists of qualitative and quantitative measures, which serve as a foundation to verify and validate the M&S. Accreditation is specific for a particular use.

2. V&V Planning

This step sets the course for the five V&V functional events: Data V&V, Concept Model Validation, Design Verification, Implementation Verification, and Results Validation. It collects and reviews formal guidance in addition to other requirements to develop constraints for the V&V effort. The plan should identify objectives, priorities, tasks, and products; allocate resources; and cover any other steps involved with verification and validation. The plan should be done in coordination with the M&S development and accreditation.

3. Data V&V

Data verification consists of ensuring that the data selected are the most applicable to meet M&S requirements. Data validation consists of ensuring that the data used accurately represents the aspects of the real world being modeled or simulated. The five data types are listed below (Figure II-3).

Data Type	Description
Reference	Descriptive information (metadata) including administrative, descriptive, and quality
Hard-wired	Data values implemented in code
Instance	Input and output data
Validation	Real world facts
Exchange	Data exchanged across a federation or distributed simulation

Figure II-3: Data Types (From Navy Modeling and Simulation Management Office, 2004)

4. Conceptual Model Validation

This step confirms that the conceptual model's capabilities meet the M&S's requirements. The conceptual model bridges the gap between defined requirements and M&S design. The main goal of this step is to demonstrate that the M&S accurately and completely represents requirements and how assumptions, limitations, and architectural structure will impact M&S use.

5. Design Verification

This step confirms that the design is true to the conceptual model. It ensures that specifications and functional designs accurately reflect the concept, meets requirements, and satisfies the acceptability criteria for accreditation.

6. Implementation Verification

This step determines that the M&S was developed correctly and works as designed. This is the documented test and review process that determines whether the M&S accurately represents the conceptual model and the given requirements. The end product is the actual model or simulation with a V&V Report documenting all uncovered flaws and their impacts.

7. Results Validation

This step determines if the developed M&S addresses the requirements for its intended use. It is the documented process that reviews the behavior of the M&S with the behaviors of the real system under test. This can take the form of output comparison, benchmarking, etc.

8. Accreditation Implementation

This step consists of multiple activities. Once the accreditation package is received, the **accreditation assessment** begins. The package is evaluated according to the Accreditation Plan and the M&S qualities are compared against the acceptability criteria. Discrepancies, workaround recommendations, remained risks, and limitations in M&S use are then identified and documented. An **accreditation decision** is then made (Full Accreditation/Limited or Conditional Accreditation/Non-accreditation) based off of the evaluation. In the event that the M&S might be reused for other applications or if the M&S is updated, **accrediting the reused M&S** then needs to occur.

E. TYPES OF V&V TECHNIQUES

There are seventy-five Modeling and Simulation V&V techniques (Table II-1) that are derived from the software engineering and M&S fields. The techniques chosen to V&V a model or simulation depends on the following criteria:

- The Simulation Type
- The Problem
- The Acceptability Criteria
- The M&S Objective and Requirements
- The User Risks and Priorities
- The Constraints and Restraints (time, money, personnel, equipment)

The online VV&A Recommended Practices Guide divided the various types of V&V into four main categories: Informal, Static, Dynamic, and Formal. The Recommended Practices Guide describes the four categories as the following:

> **Informal techniques** are among the most commonly used. They are called informal because they rely heavily on human reasoning and subjectivity without stringent mathematical formalism. The informal label should not imply, however, a lack of structure or formal guidelines in their use. In fact, these techniques should be applied using well-structured approaches under formal guidelines. They can be very effective if employed properly. (Dobey et al., 2006)
>
> **Static V&V techniques** assess the accuracy of the static model design and source code. They can reveal a variety of information about the structure of the model, modeling techniques used, data and control flows within the model, and syntactical accuracy. Static techniques do not

require machine execution of the model but mental execution or rehearsal is often involved. Static V&V techniques are widely used and many automated tools are available. For example, the simulation language compiler is itself a static V&V tool. (Dobey et al., 2006)

Dynamic V&V techniques evaluate the model based on its execution behavior and as such require model execution. Most dynamic V&V techniques require ***model instrumentation,*** the insertion of additional code (probes or stubs) into the executable model to collect information about model behavior during execution. Probe locations are determined manually or automatically based on static analysis of the model's structure. Automated instrumentation is accomplished by a preprocessor that analyzes the model's static structure (usually via graph-based analysis) and inserts probes at appropriate places. (Dobey et al., 2006)

- Dynamic V&V techniques usually are applied in three steps:

  1) Executable model is instrumented

  2) Instrumented model is executed

  3) Model output is analyzed and behavior is evaluated

Formal V&V techniques are based on formal mathematical proofs of correctness. If attainable, a formal proof of correctness is the most effective means of model V&V. Unfortunately, "if attainable" is the sticking point. Current formal proof of correctness techniques cannot even be applied to a reasonably complex simulation; however, formal techniques can serve as the foundation for other V&V techniques. (Dobey et al., 2006)

Table II-1: V&V Techniques (From Dobey et al., 2006)

Verification and Validation Technique Taxonomy			
Informal Techniques			
audit	**desk check**	**face validation**	**inspection**
review	**Turing test**	**walk-through**	
Static Techniques			
cause-effect graphing	**control analyses** calling structure; control flow concurrent process; state transition	**data analyses** data dependency; data flow	**fault/failure analysis**
interface analyses model interface; user interface	**semantic analysis**	**structural analysis**	**symbolic evaluation**
	syntax analysis	**traceability assessment**	
Dynamic Techniques			
acceptance test	**alpha test**	**assertion check**	**beta test**
bottom-up test	**comparison test**	**compliance tests** authorization; security performance; standards	**debugging**
execution tests Monitor; profile; trace	**fault / failure insertion test**	**field test**	**functional test (Black Box test)**
graphical comparison	**interface tests** data; Model; user	**object-flow test**	**partition test**
predictive validation	**product test**	**regression test**	**sensitivity analysis**
special input tests		**structural tests (White Box tests)**	**statistical techniques**
boundary value equivalence partitioning extreme input invalid input	real-time input self-driven input stress trace-driven input	branch; loop condition; path data flow; statement	**submodel / module test**
symbolic debugging	**top-down test**	**visualization / animation**	
Formal Techniques			
induction	**inference**	**logical deduction**	**inductive assertion**
lambda calculus	**predicate calculus**	**predicate transformation**	**proof of correctness**

F. ROLES AND RESPONSIBILITIES

DoD Instruction 5000.61 designated individual Components Commands (e.g., Navy) to designate, delegate authority, and assign key roles and responsibilities in the VV&A process. This means that each DoD Component will have roles assigned to different personnel in relation to its different M&S organizations (e.g., Operations and Experimentation, Training, Acquisition, and Analysis and Assessment). Multiple roles can be given to a single person or organization. Despite the differences in assignment, the roles are the same. Roles definitions are listed below (Navy Modeling and Simulation Management Office, 2004) as well as their relationships to one another:

Accreditation Agent: The individual, group, or organization designated by the Accreditation Authority to conduct an accreditation assessment for an M&S.

Accreditation Authority: The organization/individual who approves the use of an M&S for a particular application. The Accreditation Authority represents the M&S User's interests. The Accreditation Authority is a government entity.

DoD Modeling and Simulation Executive Agent (MSEA). The DoD-assigned organization with responsibility and authority for development and maintenance of a specific area of M&S application, including relevant standards and databases used by or common to many M&S capabilities.

M&S Developer: The individual, group or organization responsible for developing or modifying a simulation in accordance with a set of design requirements and specifications.

M&S Proponent: The organization that has primary responsibility for M&S planning and management that includes development, verification and validation, configuration management, maintenance, use of an M&S, and others as appropriate. The M&S Proponent is a Government entity.

M&S User: The individual, group, or organization that uses the results or products from a specific application of an M&S. The M&S User is a Government entity.

Subject Matter Expert: An individual who, by virtue of education, training, or experience, has expertise in a particular technical or operational discipline, system, or process.

Verification and Validation (V&V) Agent: The individual, group, or organization designated by the M&S Proponent to verify and validate an M&S.

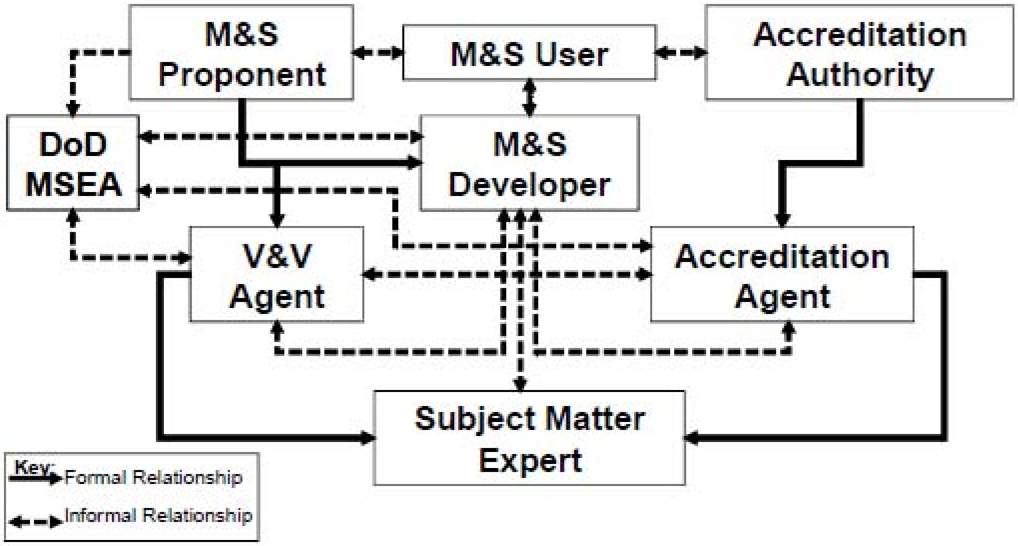

Figure II-4: Relationships between VV&A Roles (From Navy Modeling and Simulation Management Office, 2004)

Table II-2 shows a sample designation of task responsibilities to the different roles in the M&S and VV&A process. Actual designations of responsibility may vary as long as the actual process is followed.

Responsibility	M&S User	Accred. Authority	Accred. Agent	M&S Proponent	V&V Agent	M&S Developer	SME
M&S Need	Approve/ Assist	Monitor/ Review	Lead	Review	Assist	Assist	Assist
Requirements Dev & Mgmt	Approve/ Assist	Monitor/ Review	Lead	Review	Assist	Assist	Assist
Technical Solution	Assist					Perform	Assist
Product Integration	Assist					Perform	Assist
Support	Assist					Perform	Assist
Project Management	Assist					Perform	Assist
Accreditation Planning	Review	Approve	Lead	Monitor	Review	Review	Assist
V&V Planning	Review	Monitor	Review	Approve	Lead	Review	Assist
Data V&V	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Conceptual Model Validation	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Design Verification	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Implementation Verification	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Results Validation	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Accreditation Implementation	Review	Perform/ Approve	Perform/ Assist				Assist

Table II-2: Sample M&S VV&A Role Responsibility Matrix (From Navy Modeling and Simulation Management Office, 2004)

G. DOCUMENTATION

The DVDT (DoD VV&A Documentation Tool) assists users in planning, implementing, and documenting the VV&A process by automating and standardizing VV&A documentation. It helps produce the following documents (in order of development):

- **Accreditation Plan** – The documents M&S requirements, acceptability criteria, in addition to measures and metrics to be used.
- **V&V Plan** – This document interprets the accreditation plan and refines requirements. It details V&V methodology, risks, personnel, funding, and schedule.
- **V&V Report** – This presents the evidence that supports the fidelity and functionality of the model or simulation in order to meet requirements.
- **Accreditation Report** – This summarizes the assessment of the evidence produced by the V&V and documents the credibility and usability of the M&S.

Once completed and an accreditation decision made, the Accreditation Decision Letter is drafted and signed by the Accreditation Authority.

H. SUMMARY

Although not a full replacement for a real system, the advantages of using models and simulations are tangible and clear. They provide cost-effective, repeatable, controlled, safe, and rapid ways to aid in decision-making or training. They can be kept as artifacts and revisited at futures times and places as needed. Models and Simulations are key support tools for DoD programs and are thus mandated accordingly. Failure to take advantage of M&S capabilities generates enormous and unnecessary risk. The following is just a small set of dangerous scenarios when M&S is not used.

- Test pilots learn to fly a prototype aircraft without the benefit of using a simulator first.
- DoD Components testing the joint interoperability of their C4I systems after they are employed within their organizations.
- Network specialists deploying GIG untested network configurations on the production environment.
- Navy ships being constructed without models being tested against various sea states.

In essence, Models and Simulations are used to protect time, lives, and money. The Verification and Validation process helps ensure that an M&S does what is needed to be done. Accreditation gives the customer the credibility they seek to use the M&S.

III. OVERVIEW: VIRTUAL MACHINES

Virtual Machines (VM) (i.e., software) are logical implementations of physical hardware (i.e., computers). They are ideal C4I Modeling and Simulation environments because their main purpose is to simulate a physical computer as closely as possible. They do it so well that they are standard replacements for physical servers and physical workstations in production environments. At the time of this writing, all Fortune 100 companies are taking advantage of the power of virtualization (VMware, 2010b). This chapter will introduce the concept of virtualization, the different variations of virtual machines, and their advantages and disadvantages.

A. COMMON TERMINOLOGY

ABI (Application Binary Interface) - This provides the interface between an application program and hardware resources. It consists of a set of user ISAs (see ISA definition below) but does not give access to system ISAs. Instead, system calls are brokered by the operating system which manages hardware resources (Figure III-1).

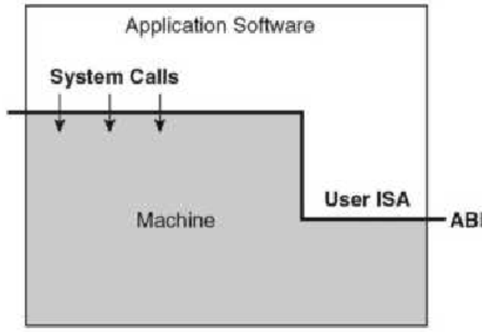

Figure III-1: ABI depiction (From Smith & Nair, 2005)

Host System - This is the underlying physical platform that the virtualization software is installed upon. It can consist of just the computer hardware or both the computer and the operating system.

ISA (Instruction Set Architecture) - This is the dividing line (and communication interface) between hardware and software. It is the instruction set that allows operating systems and/or applications to interact with the hardware. *System ISA's* are visible to operating systems and allow the managing of hardware resources. *User ISA's* are the ones visible to application programs (Figure III-2)

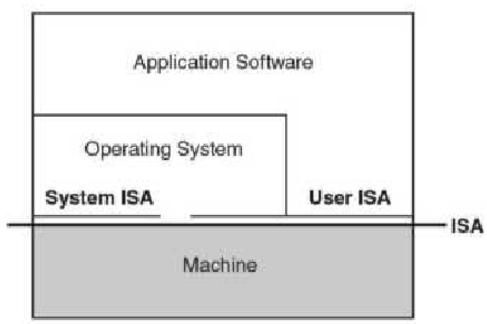

Figure III-2: ISA depiction (From Smith & Nair, 2005)

Guest - The application or operating system that runs on top of the Virtual Machine.

Runtime – The virtualization software for a process VM. It accepts ABI calls from an application and relays them to the underlying operating system and physical machine.

B. WHAT IS A VIRTUAL MACHINE?

The more common definition for a virtual machine (VM) is a software implementation of a physical machine (i.e., computer). What a virtual machine consists of is a matter of perspective. Depending on the method of categorization, VMs can be distinguished in many ways. This paper will distinguish them by using the taxonomy below (Figure III-3). The first level divides VMs on whether it virtualizes a computer system or if it virtualizes a computer system AND the operating system. The second level divides VMs on whether the Instruction Set Architecture (ISA) is the same for both the guest and the host platforms.

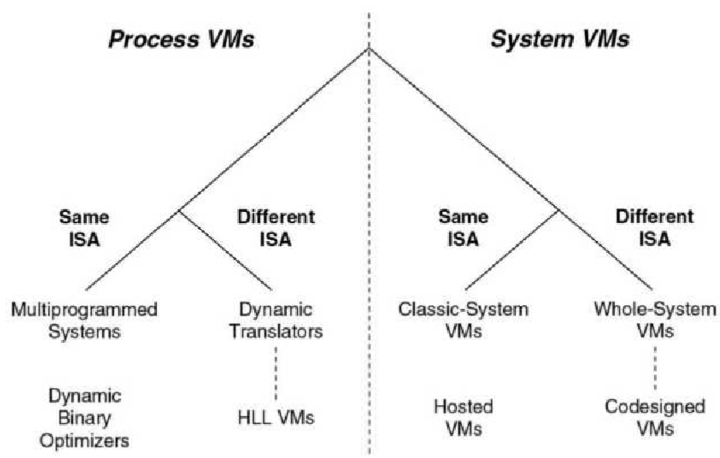

Figure III-3: VM Taxonomy (from Smith & Nair, 2005)

1. Process VMs

From the perspective of a process that is executing a computer program or application, the VM is a combination of both the operating system and the computer system. The virtualization software in a process VM is normally called the Runtime (Figure III-4). This type of VM provides a common ABI for the guest. The Runtime requires an Operating System to be already installed and serves to decouple the application from the operating system. The Operating System that the guest application thinks it is running on need not be the same operating system that is installed on the host (e.g., The guest application runs on one ABI and the host OS utilizes a different ABI).

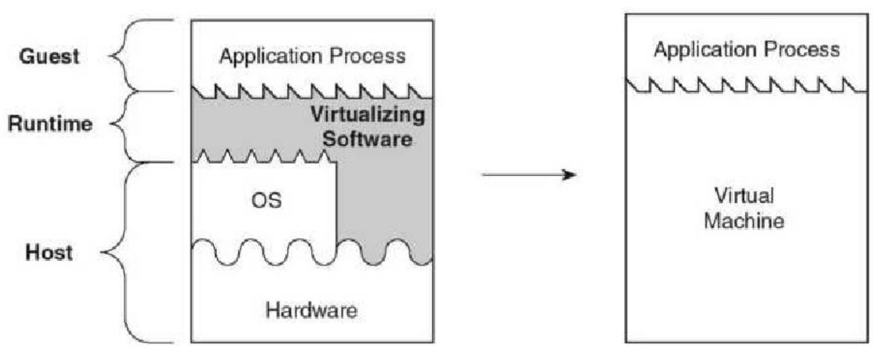

Figure III-4: Process VM (From Smith & Nair, 2005)

a. Process VMs With Same Host-Guest ISA

This type of Process VM fools an application process to think that it has a complete Operating System (that uses the same ISA) all to itself. The application still interfaces with the ABI that the Runtime provides for it but user ISA calls can be run without any form of translation or modification. Examples include

Multiprogramming, Binary Optimizers, and Agentless Application Virtualization. Multiprogramming is the allocation of resources to more than one process. Binary Optimizers perform code optimizations for an application. Agentless Application Virtualization (e.g., VMware ThinApp) is the decoupling of an application from the operating system in which OS registries or system files hold no clues to the application's existence. In all these cases, the host operating system would be in charge of managing hardware resources.

Fidelity – ABI translation/emulation may occur depending on the similarity of the host Operating System ABI and the Runtime provided ABI. A user ISA call by the guest application needs little or no translation.

Portability – One version of an application can be run on different versions of an Operating System (or even a different operating system). Different Runtime versions will be necessary however. The hardware is also limited to those that have the same ISA.

Performance – ABI translations/emulations may need to occur if the host OS ABI and the Runtime provided ABI is different. This may cause performance degradation. On the same token, performance can be increased (e.g., binary optimizers) when the ABI's are the same. ISA calls need little or no translation/emulation, which aids in performance.

Replication – Depending on the process VM implementation, multiple instances of the same application can run on a single operating system with minimal or no conflict (e.g., Agentless Application Virtualization). In such a case, the Runtime manages encapsulates each guest application in its own "instance" of an OS (with their own set of ABI).

b. Process VMs With Different Host-Guest ISA's

These VM's translate ABI calls from the guest to the host through a process called emulation. It fools the application that it is actually running on a host with the same ISA. Examples include Dynamic Translators and High Level Language Virtual Machines (e.g., Java VM). This offers a lot of flexibility for running applications but reduces performance. Example: OpenOffice running on a JVM (Java VM) which can run on Windows, Mac OSX, Solaris, etc.

Fidelity – Fidelity is low. ABI translation/emulation occurs. Any user ISA call by the guest application needs translation or emulation.

Portability – One version of an application can be run on different Operating Systems (assuming that version of the VM exists for that OS) despite the hardware it is on.

Performance – ABI translations/emulations occur which generate a large overhead. This causes performance degradation.

Replication – Depending on the process VM implementation, multiple instances of the same application can run on a single operating system with minimal or no conflict. In such a case, the Runtime manages encapsulates each guest application in its own "instance" of an OS (with their own set of ABI).

2. System Virtual Machines

From the perspective of an operating system, this VM is a software implementation of an entire computer system with a complete set of hardware resources to include CPU, Disk, I/O, Memory, etc. Servers or laptops alike can be virtualized. The VM provides a common ISA and a set of virtual hardware resources for every operating system installed on it. The virtualization software is normally called the VMM, or Virtual Machine Monitor (Figure III-5), and acts as a resource manager for the host platform. If the host platform doesn't have a physical resource for a needed virtual resource, it can emulate the desired action of the resource.

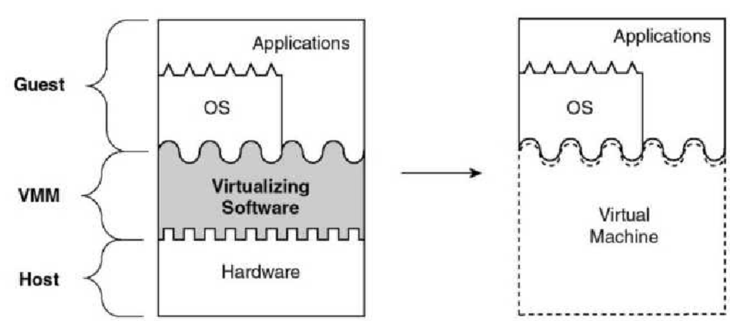

Figure III-5: System VM (From Smith & Nair, 2005)

a. *System VMs With Same Host-Guest ISAs*

In this type of system VM, the VMM (VM Monitor) acts as a hardware resource manager. Classic System VMs and Hosted VMs fit under this category. Classic System VMs have VMMs installed directly on the hardware and are more commonly known as "hypervisors" (e.g., VMware ESX Server, Citrix Xen). Hosted VM's have VMM's installed on a host OS (e.g., Parallel Desktop). Since an operating system and/or application uses the same ISA, user and system calls are not translated. For most intents and purposes, the calls are merely managed to allow a guest (or multiple guests) to think it has a computer all to itself (Figure III-6). Example: VMWare's ESX Server.

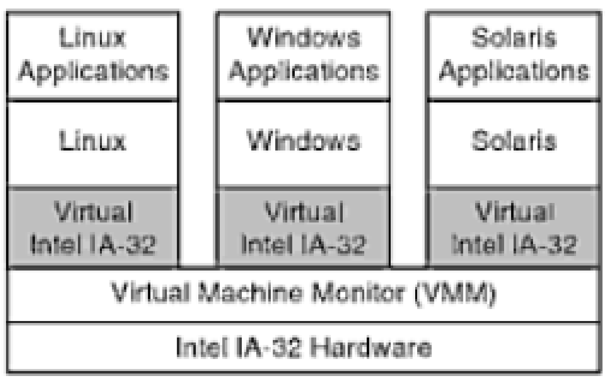

Figure III-6: Classic System VM (From Smith & Nair, 2005)

Fidelity – The VMM primarily works as a hardware resource manager. Despite this, some ISA translation/emulation occurs—even when ISAs are the same. This type of VM is the closest to the "real" machine when compared to other VM's.

Portability – As long as the ISA stays the same (e.g., the hardware is the same), the virtualized guest operating system can be moved to different

VMM's. For example, a virtualized Windows XP VM (the guest) can be moved from an Intel laptop to another VMM-enabled Intel desktop.

Performance – Performance can reach near native speeds depending on the System VM implementation since ISA translations/emulations do not need to occur.

Replication – Depending on the system VM implementation, multiple instances of the same operating system can run on a single hardware with minimal or no conflict. In such a case, the VMM encapsulates each guest OS in its own "instance" of a hardware system.

b. System VMs With Different Host-Guest ISAs

These VMs require the use of emulation (e.g., dynamic binary translation) to work. They convert system and user calls from the guest ISA to the host ISA. It fools the operating system that it is installed on hardware with the same ISA. Whole system VMs and Codesigned VMs fall under this category. The former is designed for portability of operating systems and applications. For example, Windows XP being installed on a PowerPC-based Mac OS (Figure III-7). This offers a lot of flexibility but reduces performance. Codesigned VMs, on the other hand, are designed with hardware optimization in mind. They are designed concurrently with the host hardware in order to optimize ISA translations and allow for hardware innovations.

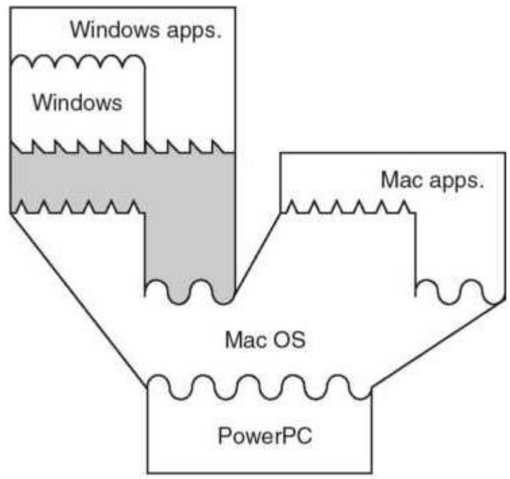

Figure III-7: Whole System VM (From Smith & Nair, 2005)

Fidelity – Whole system VM fidelity is low because ABI and ISA emulations occur. The difference between host-guest ISAs and ABIs will dictate the degree of the fidelity. On the other hand, Codesigned VMs are so closely linked to hardware development that the VM software is part of the actual hardware implementation.

Portability – Whole system VMs enjoy great hardware portability since whole operating systems can be moved from one hardware system to another. Codesigned systems, however, are tied specifically to a close family of hardware.

Performance – ABI and ISA translation causes performance degradation for whole system VMs. Codesigned VMs encounter very little performance degradation since they are developed so closely to a family of hardware and are designed specifically for ISA optimization.

Replication – Whole system VMs can be replicated easily since they are often represented as files. Codesigned VMs, however, are generally not designed for replication and rarely provide the capability.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. OVERVIEW: COMMERCIAL-OF-THE-SHELF (COTS) SYSTEMS

The buy-versus-build dilemma is a persistent question that haunts IT decision makers. This chapter aims to discuss the use of COTS systems. It will first touch on a brief background and history behind commercial-off-the-shelf software and the DoD Directives related to it. It will then discuss common assumptions surrounding COTS, as well as lessons learned when using them. Third, it introduces a risk assessment chart that will help evaluate COTS software before implementing them.

A. BACKGROUND AND HISTORY

Much like the manufacturing breakthrough of the interchangeable part, the concept of the reusable code emerged as a key goal to reduce software costs in the 1970s and 1980s. If a software architect designed software with reusability in mind, the software components could be utilized again in different parts of the program. Success in this area was limited. In the 1990s, object-oriented software made reuse more feasible and software components were easier to integrate even when developed separately. As interface standards matured, whole software system packages became easier to implement and integrate with one another. This progression of technology allowed for the use of COTS software as a means of saving resources such as time and money. Some even conclude that most developmental efforts will not be able to afford <u>not</u> to use COTS (McKinney, 2001).

The DoD defines COTS as "one that is sold, leased, or licensed to the general public; offered by a vendor trying to profit from it; supported and evolved by the vendor who retains the intellectual property rights; available in multiple, identical copies; and used without modification of the internals" (Office of the Secretary of Defense, 2000). For the purposes of this thesis, COTS software will generally be referred simply as COTS.

Understanding that the DoD no longer drives many technologies critical to military systems, the DoD issued policies to take advantage of the innovation and development in the commercial marketplace. DoD Directive 5000.1 describes the management principles that apply to DoD Acquisition programs. It states the following in regards to the use of commercial products, services, and technologies (DoD Directive 5000.1, 2003):

> In response to user requirements, priority consideration shall always be given to the most cost-effective solution over the system's life cycle. **In general, decision-makers, users, and program managers shall first consider the procurement of commercially available products, services, and technologies, or the development of dual-use technologies, to satisfy user requirements, and shall work together to modify requirements, whenever feasible, to facilitate such procurements.** [Emphasis added] Market research and analysis shall be conducted to determine the availability, suitability, operational supportability, interoperability, and ease of integration of existing commercial technologies and products and of non-developmental items prior to the commencement of a development effort.

By exploiting the innovation of the commercial market, the DoD hopes to gain the benefits of a "reduced cycle time, faster insertion of new technology, lower life cycle costs, greater reliability and availability, and support from a more robust industrial base" (Office of the Secretary of Defense, 2000). "Reduced cycle time" can be achieved by leveraging the market's innate competitive environment. A "faster insertion of new technology" can be achieved by employing an already existing system as opposed to developing a solution from inception. "Lower life cycle costs" can be achieved by leveraging the market's economies of scale in addition to the reduced need to employ and maintain resident software engineers. "Greater reliability and availability" can be achieved by leveraging the vendor's innate desire to profit and survive in a competitive market. Finally, "support from a more robust industrial base" can be achieved by leveraging the commercial industry's expertise and maturity in their given specialization.

B. SILVER BULLET OR PANDORA'S BOX?

Despite the compelling benefits that COTS software promise the DoD, it is not an end all solution to every military software problem. In fact, the benefits that organizations seek sometimes turn out to be pitfalls. For example, purchasing a first-to-market COTS from a startup company reverses the benefit of having a strong industrial support base; buying a COTS that needs extensive modification or integration negates reliability and availability; or using bug-ridden COTS nullifies the desire for reliability.

In one case study, a COTS package was purchased because it provided the best functionality on the market. Unfortunately, its first releases were full of bugs, documentation and support was poor, and it was only available in one platform. In the end, the developer started over again from scratch and had to develop an in-house solution. They wasted both resources and time (Galorath & Evans, 2006).

To avoid similar disasters, DoD Program Managers need to have realistic expectations regarding COTS and the following assumptions need to be tested and scrutinized. Assumptions that are not validated turn into risks. They can be categorized into technology, vendor, and product assumptions. Some common assumptions are listed below.

1. Technological Assumptions

- *Once adopted by the DoD, the technology will persist in the long term.* The DoD is not enough of a market driving force to keep a technology afloat. Determine market acceptance of the technology.
- *The DoD can rely on the competitive market to keep COTS innovative and inexpensive.* Emerging technologies, unprofitable technologies, or monopolized markets may keep the number of companies limited. Perform market research to understand the market space. (Hensley, 2000)
- *Personnel can be found to maintain the technology.* The market acceptance of a technology often dictates how many personnel

are trained in it. On the same token, a popular technology can create a competitive market to trained personnel.

2. Vendor Assumptions

- *Vendors can provide long term support.* Due to the competitive nature of the market, there is no guarantee that a vendor will stay in business. Examine the vendor's maturity and competitive edge. A vendor may also lack necessary support personnel. Examine their support practices.

- *System Integrators or Consultants are experts in the COTS they are integrating or implementing.* System Integrators or consultants learn technologies that bring in business which sometimes imply that they jump around from one implementation to another while learning about technologies on the fly. Examine the work experience and history of system integrators or consultants. (Galorath & Evans, 2006)

- *Vendors will keep the COTS current with innovative technologies.* A vendor may go out of business, is unable to keep up, or be a monopoly. Examine the vendor's maturity, stability, and maintenance/development practices. (Hensley, 2000)

- *The DoD can drive the development of the COTS to fit its needs.* The market ultimately drives the development. The DoD may just be another customer amongst many that is vying for COTS features. The vendor may take away essential features or add unwanted features despite DoD feedback. Examine how responsive the vendor is to customer feedback. (Office of the Secretary of Defense, 2000)

3. Product Assumptions

- *Integrating or modifying COTS will be cheaper and faster than building it yourself.* Improperly implemented, COTS can sometimes take longer or be more expensive than a custom-built system. The complexity and stability of the COTS, the skill of the system integrator, and the amount of modification/integration will dictate the time and resource costs. (Hensley, 2000)

- *Vendors will use commercially accepted interface standards.* A vendor will sometimes use proprietary interfaces to maintain market control. (Hensley, 2000)

- *COTS literature is accurate and complete.* Not all vendors are proficient in documentation and support manuals. Examine their reputation and previous documentation products. (Hensley, 2000)

- *Integrating COTS is relatively easy.* The ease of integrating COTS is dependent on its openness and library of interfaces. Examine interface libraries, documentation, and COTS reputation in integration. (Galorath & Evans, 2006)

- *<u>COTS are relatively defect-free. We only need to perform integration testing.</u>* Although vendors try to keep defects low in a vendor to stay competitive, any program can contain defects. Trust but verify. (Galorath & Evans, 2006)
- *<u>The COTS package will meet all user requirements.</u>* Systems will not always meet user requirements without modification or extension. Vendors use a set of assumptions and requirements that may not match those of the customer. (Galorath & Evans, 2006)
- *<u>COTS are priced as if they enjoy dramatic market economies of scale or due to a competitive market.</u>* Some COTS are sold to a limited customer base (e.g., DoD) or some vendors enjoy a monopoly. Understand that vendors look to gain maximum profit. (Galorath & Evans, 2006)
- *<u>The product will inherently deal with security issues.</u>* Despite the increasing adoption of security features in software, security must still be addressed. Furthermore, implemented security features may not conform to DoD security policies.

C. LESSONS LEARNED

Over time, organizations have come to understand the potential pitfalls of COTS. Guidelines and best practices have been published to help avoid them. Numerous articles have been written regarding the Buy-versus-Build dilemma (Webster, 2008) and the Office of the Secretary of Defense published a report (2000) that discussed the considerations and learned lessons from using COTS. They are categorized into three themes: Adopting Commercial Business Practices, Evaluating Software, and Working with Contractors and Vendors.

1. Adopting Commercial Business Practices

The increased reliance on COTS requires a move away from the traditional model to the recommended model illustrated in Figure IV-1. In the traditional model, the system context, architecture, and design drove implementation. The recommended model reveals the reality that the marketplace needs to influence the system context, architecture, and design (even requirements) to maximize the benefits received. It is a model of cooperation rather than one of forced structure. Often, a program manager is just another buyer and must conform to marketplace norms to influence COTS development. Table IV-1 is the summary of suggestions to help embrace commercial business practices taken from The Office of the Secretary of Defense report.

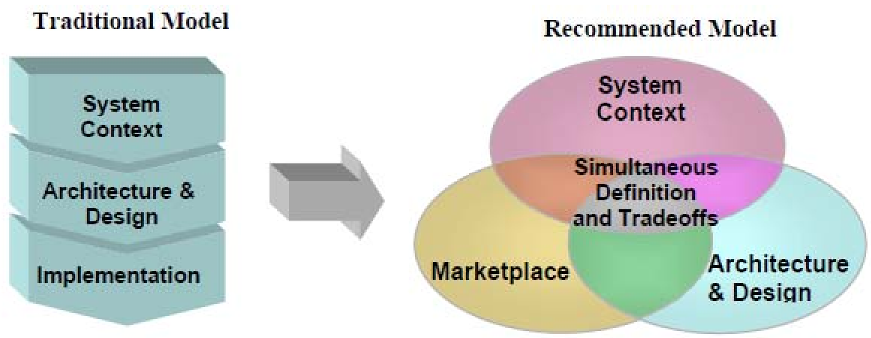

Figure IV-1: Recommended Acquisition Paradigm Shift (From Office of the Secretary of Defense, 2000)

As implied in Figure IV-1, the program manager does not drive COTS development. Whereas the System Context and Architecture previously determined implementation, they must now overlap with Marketplace needs and requirements. The vendor, based on its perceptions of product profitability, will determine performance and functional features and enhancements. This is an advantage in that the program officer does not need to directly fund features and enhancements. It can be a disadvantage in that needed features are removed or unnecessary features are added.

This means though that there will be a gap between the DoD system context and commercial use of the COTS. These gaps must be identified and bridged through investigation and negotiation. Requirement specifications must be flexible and negotiable. Compromises and the desire to bridge the gap should not surpass DoD standards and compliance documents however. One must understand which requirements are firm and which are negotiable. If the gap is too large or non-negotiable, a COTS solution is not be appropriate.

Another lesson learned is that modifying COTS is not a best practice for reconciling COTS use and program requirements. Cost and schedule overruns are common depending on the scale of the modification and complexity of the COTS. Modifications also negate the COTS benefit of outsourcing upgrades to the vendor. Future versions of the COTS may not work with the modifications and maintenance personnel will be required to upkeep the COTS—effectively making it a custom-build system.

Finally, stake-holder buy-in is essential before employing COTS. Given that COTS acquisition, implementation, and use tend to introduce changes in organization and process, it is important to involve key stakeholders early in the process. These stakeholders are often the tipping point between success and failure. Therefore, they need a clear understanding of what they are being offered.

Table IV-1: Suggestions for Embracing Commercial Business Practices

To understand the marketplace
Conduct market research independent of the contractor.
Identify all significant commercial players in the relevant application area.
Participate in the relevant conferences, trade shows, and user, professional, and standards groups.
Identify the technology domains represented by the application area.
To understand the system context
Track changes to all commercial item guidelines and direction from the DoD.
Reference these guidelines and direction in contract specifications.
Propose changes to guidelines and direction to reflect new commercial items needed in the system context.
Maintain a flexible view of requirements and business practices.
Identify all of the stakeholders and involve them early.
Pare down stated requirements to reflect only essential stakeholder needs.
To bridge the gap
Determine the gap between the capabilities and services provided in the marketplace and those required by the system.
Include the vendor in tradeoff discussions when possible.
Provide incentives to encourage the contractor to investigate all solutions that lead to the appropriate outcome.
Don't modify the commercial item.
Plan for a life-cycle support system for any modified commercial item.
Plan to make repeated tradeoffs among the system context, the architecture and design, and the capabilities in the marketplace.
Document all tradeoffs made.
Provide early functional demonstrations to get stakeholder buy-in.

2. Evaluating the Software

Before implementing a COTS solution, it must be fully evaluated. The definition of "evaluation" used here is broader than the source selection criteria used in Acquisition circles. The Office of the Secretary of Defense report (2000) broadens the language to cover the identification of commercial capabilities to help define source selection criteria, choose alternate architectures and designs, determine if future release will meet requirements, and ensure that the commercial item functions as expected. Table IV-2 is the summary of suggestions for evaluating COTS taken from the Office of Secretary of Defense report.

Characteristics such as security and information assurance, inter-operability, reliability, and maintainability are of particular importance to the DoD. Evaluators need to remember that COTS tend to be geared towards commercial users and their characteristics and requirements don't always conform to DoD regulations and needs. One also needs to realize that evaluating COTS may mean comparing solutions that don't compare well. Vendors often use different assumptions about the COTS and how it would be used. When evaluating COTS against each other, one must first decide on the system architectures that best reflect the best use of the COTS.

Another major lesson learned is that commercial items are not always commercial-off-the-shelf. One-of-a-kind systems with no market base (e.g., DoD specific systems) or systems that need modifications in order to work nullify the benefits the DoD seek in using COTS. One-of-a-kind systems lose the benefit of market competition to drive innovation up and bring costs down. Modified COTS have reduced maintainability, upgradability, and implementation advantages than their unmodified counterparts.

Finally, test beds and continued evaluations were important lessons to be learned. Vendors often do not reveal detailed information about the COTS. This limits the ability to evaluate them. In addition, new versions of the software can change rapidly. In order to ensure that the COTS still fits the needs of the DoD program, regular evaluations (formal or informal) should be made.

Table IV-2: Suggestions for Evaluating Software

To develop the skills needed
Employ outside experts to support program-office evaluation activities.
Train the program office and the stakeholders on how to evaluate commercial items.
Repeat this training as personnel or the nature of the commercial items being evaluated change.
Select a contractor who has past experience in evaluating commercial items.
To conduct evaluations
Decide in advance what information you want to gain from the evaluation of a commercial item.
Select evaluation techniques based on the type of information required and the importance of the selection to the program.
Unless it is impractical, evaluate potential commercial items in a system test bed.
Consider both the capabilities of the commercial item and the business practices of the vendor.
Take into account the business motivations of the vendors.
Understand the vendor's strategy, and talk to other buyers.
Understand where you stand in relation to the vendor's other customers.
Budget for repeated evaluations throughout the program's life cycle.
To develop the skills needed
Employ outside experts to support program-office evaluation activities.
Train the program office and the stakeholders on how to evaluate commercial items.

3. Working with Contractors and Vendors

The last major set of learned lessons is categorized under contractor and vendor relationships. The Office of the Secretary of Defense report (Office of the Secretary of Defense, 2000) found that DoD programs were most effective when they adopted practices and expectations familiar to commercial vendors. This implies that the DoD should adopt commercial buying practices and be careful of underestimating or overestimating DoD’s influence on a vendor. Table IV-3 is the summary of suggestions in improving vendor relationships taken from the report.

Because vendors are often unfamiliar with DoD acquisition processes and worry about a market larger than the DoD, it is important to adopt commercial buying practices. For instance, vendor price models are often incompatible with DoD cost models, which often consist of labor hours, material, and profit. COTS are determined by other marketplace factors. The DoD needs to learn about the marketplace and must understand that they are just another buyer, albeit one who compares to large corporations.

On the same token, the DoD needs to be careful about underestimating or overestimating its influence on a vendor. Influence can be underestimated when a vendor is eyeing a DoD contract as a large profit base. The DoD can unwittingly pressure a vendor to make large modifications on their product and only require a few licenses. Conversely, one would quickly realize that DoD-specific requirements might not influence COTS (such as Microsoft Office) development if it did not provide benefit for the larger consumer market.

Table IV-3: Suggestions for Improving Vendor Relations

To adjust buying practices
Train financial management and contract personnel in commercial buying practices.
Adapt business and engineering models and acquisition strategies to accommodate the impact of using commercial items.
To develop and execute program budgets
Base planning on total ownership cost rather than catalog price.
Investigate emerging price and cost models.
Perform market research to support determinations of reasonable value.
Include a budget and schedule for unexpected commercial impact.
To strengthen program, contractor, and vendor relationships
Use contract incentives to encourage appropriate relationships.
Maintain close relationships with vendors to exploit improvements and avoid surprises.
Verify the claims made for commercial items by vendors and contractors.
Verify the availability of commercial items.
Examine any acquisition strategy to see where it can be made more flexible or better suited to the unique commercial aspects of the system in question.

D. RISK ASSESSMENT

To help avoid the pitfalls of false assumptions and to apply learned lessons, an NPS thesis by Barry Hensley (Hensley, 2000) introduces a risk assessment chart for COTS (Figure IV-2). It provides a quick overall evaluation by covering the three major risk categories of technology, product, and vendor. It asks whether a technology is mature or stable and asks whether there is competition in the marketplace. It examines maturity or stability of a vendor, their level of expertise in the technology, their responsiveness, and the quality of its technical support. Finally, it scrutinizes the market acceptance of the COTS, its stability, the openness of its interfaces, its complexity, its security features, its safety, and its documentation.

RISK ASSESSMENT CHART

Product Name/Version:

Assessment Date:

Assessed By:

Risk Category	Risk Factor	Risk Cues: Low	Medium	High	Rating
Technology	Maturity/Stability	Widely accepted technology.	Competing technologies.	Emerging technology.	
	Competition	Large number of competing products within the selected technology.	Limited number of competing products within the selected technology.	Small number of competing products or no competition within the selected technology.	
Vendor	Maturity/Stability	Large company. Applies commercially accepted development practices.	Medium company. Applies a mix of commercially accepted and ad-hoc development practices.	Small/emerging company. Applies ad-hoc development practices.	
	Technology Expertise	Maintains personnel base with expertise in the technology.	Access to personnel with technology expertise. Moving into an emerging technology.	Limited or no access to personnel with technology expertise.	
	Responsiveness	Accepts/processes customer feedback. Provides advance notice of product changes.	Accepts/processes market feedback. Provides limited notice of product changes.	Does not accept/process customer feedback. Provides no notice of product changes.	
	Technical Support	Maintains knowledgeable technical support staff. Maintains 24/7 help desk. Easy access to help desk. Easy access to patches.	Maintains semi-knowledgeable technical support staff. Restricted help desk availability. Limited avenues to access help desk. Limited access to patches.	Knowledgeable technical assistance staff not available. No help desk. No access to patches.	
Product	Market Acceptance	Wide market acceptance. Large market share. Product drives the market.	Limited market acceptance. Medium market share.	Product not widely accepted by the market. Small market share.	
	Stability/Robustness	Very few significant upgrades. No significant bugs or limited insignificant bugs.	Moderate number of product upgrades/patches. Tolerable bugs (non-critical).	Significant number of product upgrades/patches. Significant or intolerable bugs.	
	Interfaces	Uses commercially accepted interfaces. Interface documentation is available.	Uses a mix of commercially accepted interfaces and nonstandard or proprietary interfaces. Limited interface documentation.	Uses nonstandard or proprietary interfaces. No interface documentation.	
	Complexity/Features	Easy to use. Easy to install and configure. Few extraneous capabilities. No undesirable features.	Moderately easy to use. Moderately easy to install or configure. Some extraneous capabilities. May have an undesirable feature.	Hard to use. Difficult to install or configure. Large number of extraneous capabilities. Exhibits undesirable features.	
	Security	No significant security issues. No insignificant security issues.	No significant security issues. A few insignificant security issues.	Significant security issues. Many insignificant security issues.	
	Safety	No safety issues.	N/A	Safety issue.	
	Documentation	Understandable, complete, and accurate documentation package.	Acceptable documentation package. Falls short in some areas.	Poor documentation package.	
	Cost	Competitive product cost. Good warranty. Reasonable maintenance fees.	Inflated product cost. Poor warranty. Inflated maintenance fees.	Unreasonable product cost. No warranty. Unreasonable maintenance fees.	

NOTES:

Figure IV-2: Risk Assessment Chart (From Hensley, 2000)

THIS PAGE INTENTIONALLY LEFT BLANK

V. ANALYSIS AND APPLICATION

This chapter will present the DoD organizations that would benefit from C4I Modeling and Simulation via Virtual Machines. It will then present a series of recommendations that will build upon each other. For one, this thesis recommends the use of Virtual Machines. It will then recommend the use of a COTS VM implementation. Third, it will present the case for using a Classic-System Virtual machine. Next, it will present the case of performing VV&A on a modularized COTS VM. It will then recommend functional testing as the V&V method to provide credibility for the virtual machine. Finally, it will advocate a process for DCSCs that encapsulates aforementioned recommendations.

A. THE PROBLEM SET: DOD C4I SUPPORT CENTERS (DCSC)

As with any proposal, it is always important to understand the context in which a solution will operate. Therefore, before we combine the concepts of virtualization as well as modeling and simulation, it is important to understand DoD C4I Support Centers (DCSC). This section will introduce the basic concepts and the communities they support. It will then provide a representative list of DCSCs, their pertinent information, and their mission focus. Finally, it will also explain a problem trend they are facing: an increasing total cost of ownership to perform a growing mission focus given a limited budget.

1. What Are DCSCs?

For the purposes of this thesis, a DCSC is a general term used to describe organizations or entities that focus on supporting C4I applications for various DoD activities. Often part of a larger organization with a larger focus, DCSCs serve to provide C4I M&S capabilities for various communities such as Analysis, Acquisition, Experimentation, Training, Planning/Operations, and Testing. For example, a training command may have a technical division that sets up a classroom (i.e., model) network of C4I applications for students to learn from. Or, an organization that deals with C4I acquisitions may employ an independent group to run integration and interoperability tests on new C4I systems (using a representative model of a C4I architecture in the DoD).

2. Representative List of DCSCs

The following is a representative list of DCSCs that exist throughout the Department of Defense. Recalling that a model is a "physical, mathematical, or otherwise logical representation of a system, entity, phenomenon or process" (Sanders & Office of the Undersecretary of Defense (A&T) Washington DC, 1996), these organizations set up representative models of C4I networks to support their various communities. Not all will attempt to mirror actual DoD networks and the level of fidelity will be based on mission and need.

Joint Interoperability Test Command

Mission: JITC provides a full-range of agile and cost-effective test, evaluation, and certification services to support rapid acquisition and fielding of global net-centric warfighting capabilities.

Website:

http://jitc.fhu.disa.mil/mission.html

Marine Corps Tactical Systems Support Activity (MCTSSA)

Mission: MCTSSA will provide Marine Air Ground Task Force (MAGTF)/Joint C4I system and system of systems technical expertise and support throughout all acquisition lifecycle phases in order to ensure C4I systems are engineered, tested, certified and supported, thus enabling Marines to continue to win battles.

Website:

http://www.mctssa.usmc.mil

U.S. Army Information Systems Engineering Command

Mission: Provide systems engineering, installation, integration, implementation, and evaluation support for communications and information technology systems worldwide providing capabilities to Army Organizations, Combatant Commanders, DoD agencies, and Federal agencies in support of the Warfighter.

Website:

http://www.hqisec.army.mil/index.asp

Air Force 46th Test Squadron

Mission: The 46 TW Executes Developmental Test and Evaluation Enabling the Warfighter to put Weapons on Target in all Battlespace Media.

Website:

http://www.eglin.af.mil/units/46thtestwing/index.asp

Space and Naval Warfare Systems Command (SPAWAR)

Mission: Team SPAWAR acquires, develops, delivers and sustains decision superiority for the warfighter at the right time and for the right cost.

Website:
http://enterprise.spawar.navy.mil/body.cfm?type=c&category=38&subcat=180

3. What Problems Do They Face?

Due to the growing demand for and the increasing amount of C4I systems, DCSCs are forced to maintain rooms or facilities that model C4I networks. The community a DCSC supports determines the scale and fidelity of the C4I model. A training command may only need a classroom with networked C4I systems (i.e., as a training model) to train Operations staff while an interoperability test center may have a network of disparate DoD organizations creating joint C4I network model (i.e., as a testing model). For example, MCTSSA established the "VII MEF (Marine Expeditionary Force)" which represents a MEF C4I architecture for systems integration testing. It boasts data networking, voice switching, and multiplexing capabilities normally found in a normal MEF. Unfortunately, given the requirement to have such facilities, DCSCs are burdened with constant acquisition and maintenance costs. If these centers are fortunate, Program Offices carry acquisition costs of the actual test or training systems as part of the budget. Despite the organization that pays for them, however, the American taxpayer still pays the bill.

Unfortunately, DCSCs still have to shoulder the costs of maintenance, energy, cooling, manpower, infrastructure, and storage costs inherent to maintaining C4I environment models. These encompass the greater bulk of total costs of ownership (Figure II-1). To aggravate the issue, DCSCs that test C4I interoperability are required to maintain legacy, current, and future C4I systems (hardware and software). They need to ensure that that all versions can interoperate with the each other with little or no effect. This introduces additional burdens on facilities, manpower, and time to configure/reconfigure environments. As a simplified example, a network of three different versions of three different C4I applications installed on three different operating systems on three different hardware platforms would require 81 different configurations! (3 versions x 3 C4I applications x 3 OSes x 3 hardware platforms = 81 different configurations) Unfortunately, DCSCs have very limited budgets and cannot effectively meet their missions with such growing total costs of ownership. Its current methods for modeling C4I systems need to be rethought if they aim to meet their goals.

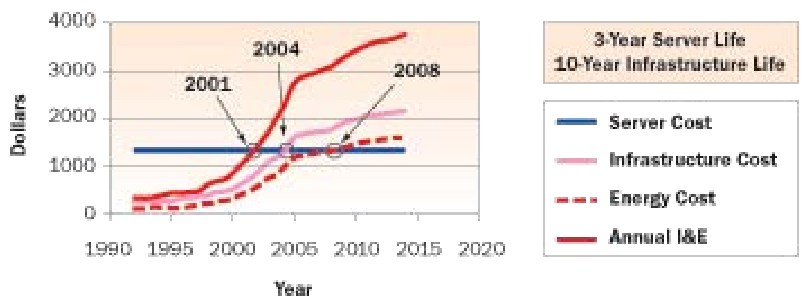

Figure V-1: Annual Amortized Costs in the Data Center for a 1U server. (From Belady, 2007)

B. THE CASE FOR VIRTUAL MACHINES

Given that many DCSCs already model C4I environments, this thesis proposes a more efficient, cost-effective, and scalable alternative. Instead of maintaining a costly hardware C4I infrastructure, one can leverage the advantages of virtual machines. The obvious next questions should then be: When should an organization use virtual machines as C4I models? And if so, what type/s should it use? Ultimately, it depends on organizational needs. An organization that places a heavy importance on platform independence because they are short on hardware resources may employ whole-system VMs. An organization seeking to have applications that are easily ported to different operating system environments may seek to leverage process VMs. Irrespective of the need, VMs in general are ideal environments for C4I modeling. Recalling the reasons for using models and simulations (see II.B), VMs provide repeatable, controlled, safe, and inexpensive environments for C4I applications.

By mere virtue of VMs being software implementations, their capability for repeatability surpasses that of hardware. Most VM technologies store their virtual machines as modularized files. Replicating them is sometimes as easy as a copy-and-paste. For example, Microsoft's Hyper-V is a system VM and stores virtual machines as a set of files. They can be cloned multiple times to create multiple copies of the same virtual machine. Some VM implementations even provide a template paradigm to facilitate configuration management.

Partially due to their implementation as software, VMs can also provide controlled environments. They allow a tester, instructor, or modeler to create a C4I environment with settings based on their needs. More advanced VMs allow for the modification of CPU speeds, RAM sizes, etc. as configuration settings in a graphical user interface (Figure V-2). This provides the user the power to give, modify, or remove hardware resources available to a C4I environment without physically opening a computer.

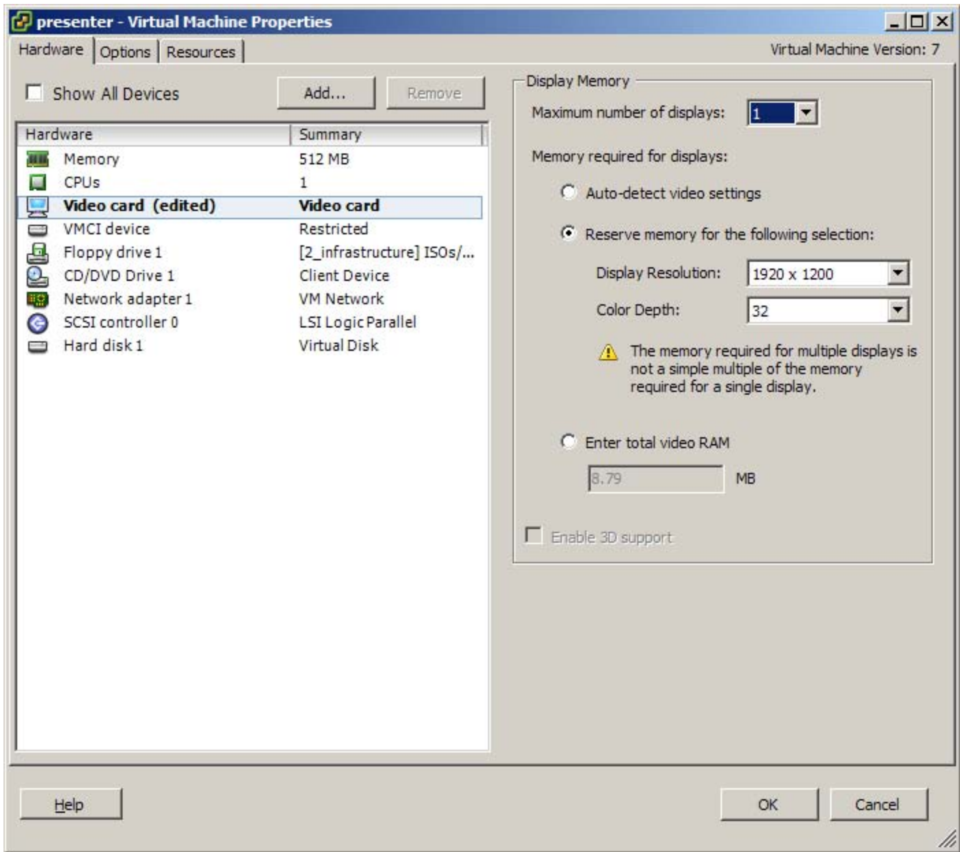

Figure V-2: VMware VSphere VM property page

Virtual environments also provide safe C4I environments. Although the "safe" characteristic is normally applied to human safety, the safety of a production environment or a physical computer is relevant as well. For example, if one needs to examine the damage of a virus in a network or the effects of an untested network configuration, a network of VMs would be an ideal environment as opposed to using it on a production environment. An infected machine could be migrated to a fenced off network segment for analysis, repair, and/or testing.

Of all the reasons organizations employ virtual environments; cost is the single largest influence. VMs are traditionally less expensive than their hardware counterparts. The cost savings can be categorized under acquisition, maintenance, infrastructure, and manpower.

- **Hardware Acquisition.** Purchasing a physical computer with its assortment of hard drives, RAM, CPU, networking equipment would be replaced by (cheaper) software licensing costs. An organization would no longer purchase multiple computers to simulate a networked environment. Instead, virtual machines are reproduced as needed on fewer machines and are only limited by licensing costs which pale in comparison to hardware purchasing costs. In addition, an organization will no longer deal with the overhead of an acquisition cycle such as buyer competition, component shortages, or delays in shipment
- **Maintenance.** The maintenance costs also fall as hardware failures decrease. Ill acting virtual machines need only to be erased and replaced by another. Alternatively, it can be send to a sandbox environment for analysis. The only hardware maintenance costs that will exist will be for the host machines that run the virtual machines. The reduction of physical computers also reduces cooling and energy costs to keep them running. These energy and cooling costs greatly exceed acquisition costs (Figure V-1). Since multiple virtual machines can reside in a single host computer, less physical machines need to be powered and less cooling would be required. In one real world instance, the

process of virtualizing 10 physical servers netted a 25% savings in energy consumption alone (Connor, May 15, 2008).

- **Infrastructure.** Physical storage also becomes a relative non-issue. Physical computers take up physical space so multiple hardware configurations will progressively take up more and more room. Since virtual machines are merely files, the number of test machines that can be stored will only be limited by the amount of hard drive space an organization can purchase. Old hardware configurations used for testing backward compatibility will no longer collect dust and valuable storage space. They will be stored as ones and zeroes in a hard drive patiently waiting for a test to need it.
- **Manpower.** When it comes to configuration, a virtual machine also excels versus its physical counterparts since many VMs are highly replicable. In order to test a particular configuration in the physical world, personnel would have to physically move and connect machines in addition to installing operating systems and applications. This process would have to be repeated multiple times depending on requirements, which often take days or weeks. In a virtual machine environment, a single VM can be configured with the desired OS and application. Multiple instances of the VM can then be generated within minutes.

With all these advantages, a virtual machine does have a disadvantage: It is still not a real machine. If a test engineer needed to test application compatibility with a specific computer (e.g., Dell Inspiron 1545, Dell PowerEdge T105 Server) or other specific hardware (e.g., Linksys Etherfast NIC, Netgear RangeMax Wireless Card), a virtual machine will not fulfill his/her requirements. Such tests can only be run on the actual hardware itself due to firmware nuances or hardware specificities. Performing a hardware compatibility test still requires acquisition of the actual hardware itself. Fortunately, this drawback only affects a limited subset of users. Primarily, developers or DCSCs that support acquisition communities test against hardware compatibility. Even in these communities, such tests are a fraction of what they perform. Interoperability, functionality, and integration tests are more common.

Irrespective of a DCSC's need, VMs are generally ideal C4I environments. They provide repeatable, controlled, safe, and inexpensive environments. When factoring in the reality that DCSCs are support organizations and are often less prioritized than deployable combat units, the allure of cost-savings coupled with greater capability is (or should be) strong. Despite the limitation that a VM does not have the fidelity of a real computer, organizations have come to realize the compelling need to use virtual machines. Software developers and enterprise-level companies routinely use virtual machines to develop and test applications before using them in production environments. Many have taken it a step further and use virtual machines as production environments.

C. THE CASE FOR COTS VIRTUAL MACHINES

Weighing the risks and benefits of COTS solutions, the implementation of commercial-off-the-shelf VMs is a sound strategy compared to developing a custom VM environment. This conclusion can be reached due to the support of (1) market competition, (2) technology maturity, (3) availability of customer support, and (4) availability for training opportunities. These strengths address the majority of risks listed in the Risk Assessment Form (Figure IV-2). They also allow DCSCs to simplify the M&S process thus furthering the case for using COTS Virtual Machines in a Modeling and Simulation environment.

Market competition in a technology is a great incentive for vendors to innovate while reducing costs for their COTS solutions. With the virtualization market projected to hit $11.7 billion in 2011, market competition will be healthy in the foreseeable future (Mann, 2007). For example: Microsoft Hyper-V, VMware VSphere, and Sun xVM are some of the competing products in the hypervisor (i.e., classic system virtual machine monitor) market alone. Despite VMware’s current leadership in the virtualization market, technological giants such as Microsoft and Sun are motivations for VMware to continue innovating while keeping costs competitive. For instance, VMware’s VSphere (then VMware ESX) was conceived in 2001 and has since gone through four major upgrades within 8 years. (VMware, 2010b)

In the context of technology maturity, virtualization is a technology developed in the 1960’s to better utilize mainframe hardware utilization originating with the IBM System 360 and 370. Although abandoned soon afterwards due to the boom in client-server technology, it regained steam in the late 1990's after underutilization became an issue again. With VMware alone boasting over 170,000 customers using their virtualization products, the market has tested and accepted the technology. As mentioned in the previous paragraph, VM innovation continues to accelerate due to competition and customer demand. COTS VMs have progressed past the basic functions of virtualization (e.g., hardware resource partitioning) and have progressed to advanced features such as disaster recovery, hardware cloning, load balancing, host clustering, central management, movement of running VMs, etc. These features contribute desirable benefits such as reduced downtime, friendlier user interfaces, remote management, role management, and decreased provisioning time. The tables found in Appendices A and B are examples of current features being touted by major vendors.

These two factors (i.e., market competition and technology maturity) contribute to a number of benefits that are attractive to DCSCs. To list a few, they motivate:

- Mature and stable virtual machines
- A strong personnel technical expertise base
- Increased training capabilities
- Responsive customer feedback and technical support
- Increased number of features/advancements
- Better documentation

In addition, they also simplify the M&S Process. In the eyes of DCSCs, their M&S need is to create a networked computer environment that closely resembles that of the real production environment. Since this need mirrors that of the COTS VM market, DCSCs benefit by using the VM vendor’s product life cycle as part of their M&S cycle. This includes developing and managing requirements, implementing the technical solution, integrating the product, providing support, and providing the overall project management for the VM (Figure V-2).

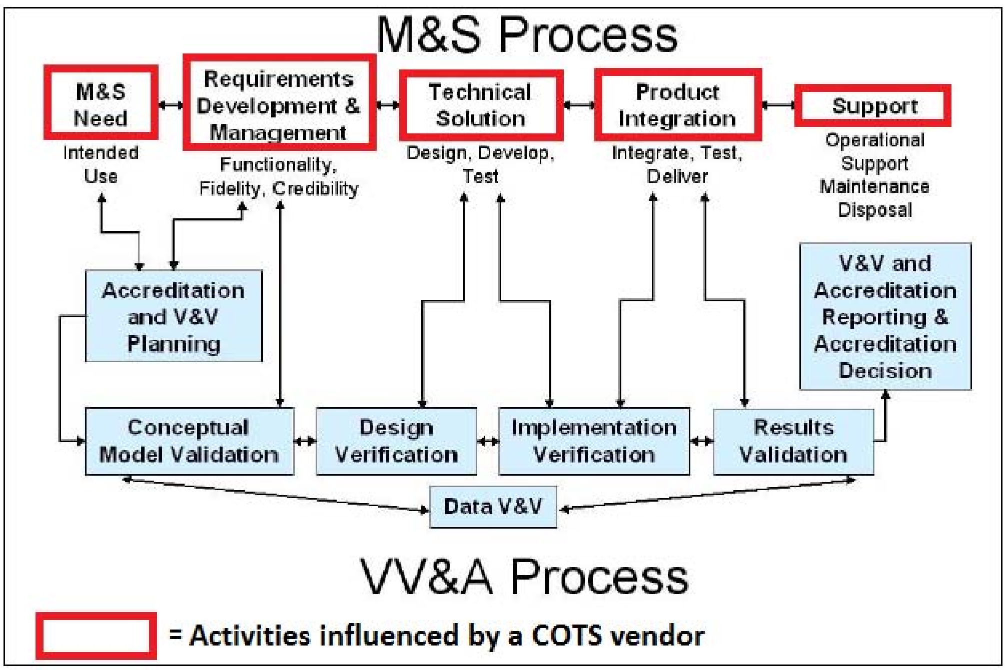

Figure V-3: COTS influence to M&S and VV&A Process (After Navy Modeling and Simulation Management Office, 2004)

It is in the vendors’ best interest to create VMs that act and operate as closely to a real system in order to convince potential customers that their VMs are at least as good as their hardware counterparts. The parallel in needs also serves to outsource the M&S Developer role from the VV&A Process (Table V-1) further streamlining the DoD activity.

Table V-1:COTS Vendor Roles (yellow) in VV&A Process (From Navy Modeling and Simulation Management Office, 2004)

Responsibility	M&S User	Accred. Authority	Accred. Agent	M&S Proponent	V&V Agent	M&S Developer	SME
M&S Need	Approve/ Assist	Monitor/ Review	Lead	Review	Assist	Perform	Assist
Requirements Dev & Mgmt	Approve/ Assist	Monitor/ Review	Lead	Review	Assist	Perform	Assist
Technical Solution	Assist					Perform	Assist
Product Integration	Assist					Perform	Assist
Support	Assist					Perform	Assist
Project Management	Assist					Perform	Assist
Accreditation Planning	Review	Approve	Lead	Monitor	Review	Assist	Assist
V&V Planning	Review	Monitor	Review	Approve	Lead	Assist	Assist
Data V&V	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Conceptual Model Validation	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Design Verification	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Implementation Verification	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Results Validation	Review	Monitor	Monitor	Approve	Lead	Assist	Assist
Accreditation Implementation	Review	Perform/ Approve	Perform / Assist				Assist

Despite all the listed advantages, disadvantages do exist. For one, subject matter experts must be trained in the vendor-specific VM. Personnel must be trained, maintained, or outsourced in order to use the COTS VM. Given the maturity of the market, however, subject matter experts are easier to acquire due to quantity and availability of industry supported and accredited training paths—much easier than if a custom technology was developed. Secondly, non-standard C4I hardware devices (e.g., cryptographic device) are unlikely to have virtualized solutions due to specific machine requirements and custom drivers written for virtual machines. Fortunately, non-standard hardware is not always necessary for many modeling applications (e.g., Cryptographic gear isn't necessary in a training environment). For those that do require the inclusion of non-standard C4I hardware, hybrid virtual-real C4I network environments are still viable and cost effective models for DCSCs. Alternatively, many COTS solutions allow for custom device drivers to be developed for virtual machines.

The use of a commercial-off-the-shelf solution for virtual environments is an obvious answer to those considering the technology. The marketplace is healthy with competition and the technology is mature despite its relative newness to many end users. The COTS implementation also allows DCSCs to outsource the M&S process as well as the M&S role in the VV&A process. Leveraging the vendor's expertise and production capabilities in this arena allows DCSCs to focus on the important technologies and implementations specific to their needs. Although manpower overhead and the inability to virtualize non-standard C4I hardware are disadvantages, they are not overwhelming hurdles and the advantages of COTS virtualization easily make up for any shortfalls.

D. THE CASE FOR THE CLASSIC SYSTEM VM

Given the multitude of virtual machine technologies highlighted in Chapter III.B, a DCSC must decide which one to use for their C4I model environments. Although the proper response is “it depends”, this thesis recommends the use of a classic-system virtual machine (Chapter III.B.2.a). This recommendation weighs in multiple factors such as a mature classic-system VM market, increased hardware capability, fidelity requirements, repeatability advantages, and the benefits of technology standardization. Classic-system VMs, however, have the disadvantage of decreased portability. This decreases its ability to be migrated to hardware with differing ISAs.

Although the maturity of virtualization technology was already discussed, the classic-system (or hypervisor) VM market is arguably the most mature. First developed as a mainframe technology in 1966, current products run on commodity hardware such as x86/x64 processors commonly found in consumer computers. Microsoft has even made its hypervisor technology (Hyper-V) as a component of its Windows Server 2008 R2 (Yegulalp, 2010). With technology giants such as Microsoft, Intel, AMD, IBM, Sun, Citrix, and VMware involved, classic system VMs cannot be designated as an emerging or niche market. This helps ensure that the technology will continue to innovate, be competitively priced, and have long-term outlook.

Partially responsible for the resurgence of virtualization was that many organizations found that increased hardware capability led to underutilization and unnecessary costs in power, cooling, and hardware in their expanding data centers. Hypervisors allowed multiple operating systems to run on a single hardware platform. With hypervisors steadily progressing as a mainstream technology, current Intel and AMD processors have added instruction sets to allow hypervisors to run more natively (and thus efficiently). These advances in hardware make classic-system VMs more stable, efficient, and capable.

The fidelity of a classic system VM compared to the hardware platform is also an advantage. It provides great fidelity compared to other virtualization methods since ISA calls stay the same and are often not emulated (Paragraph III.B.2.a). Combined with computer processor advancements, classic system VMs run more natively and efficiently on the hardware they are installed on. This level of fidelity is important to many DCSCs such as testing or acquisition organizations. Although still not a replacement for a physical C4I modeled environment, the fidelity is sufficient for tests that require large-scale C4I test environments (e.g., System of System Tests) --- the environments that produce the most overhead in time, money, and personnel.

The repeatability trait of Classic System VMs is also of great interest to DCSCs. The ability to clone multiple VMs in a few minutes would replace the time-consuming and expensive configuration of multiple computers needed to create a modeled C4I environment. Students improperly configuring a C4I application can be given a new VM environment to start anew. Analysts simulating the effects of a virus can be given a contained networked C4I environment with minimal hardware requirements since multiple VMs can run on a single machine.

Finally, standardizing on a single VM technology (i.e., Classic-System VM) leverages economies of scale. Even if different DCSCs don't need the level of fidelity required by test or acquisition organizations, standardizing on a single technology alleviates personnel expertise needs, reduces the number of contracts required to support virtualization, and is able to enjoy the benefits of purchasing enterprise level licenses. The fidelity required by test and acquisition DCSCs serves as the lowest common denominator thus making hypervisors the ideal technology to invest in.

Classic-system VMs suffer in the area of portability however. Although classic-system VMs can still be easily moved from one family of hardware to another, they are difficult to migrate to hardware with disparate instruction set architectures (ISA). The most common and mature classic-system VMs run on x86 hardware (i.e., Intel and AMD processors). Operating Systems designed for other hardware (e.g., has a different ISA such as PowerPC) are not compatible without other forms of emulation. Despite this, the portability limitation is not a large disadvantage. Given the prevalence of x86 hardware as a near de facto standard in DoD communities, the limitation is of little concern to DCSCs.

Given the above factors, classic system VMs are the most logical virtualization technology for use by DCSCs. It has the right balance of fidelity, portability, and repeatability needed. DCSCs can take advantage of the economies of scale by standardizing on a single technology and can leverage the benefits of a mature technology market. Choosing a different virtualization technology such as process VMs or whole-system VMs risks failing fidelity requirements needed by certain DCSCs. If this occurs, DCSCs will not be able to exploit the advantages of shared technology such as economies of scale.

E. THE CASE FOR ACCREDITING A REUSABLE VM MODULE

The next step in solving the problems that DCSCs encounter is to modularize their M&S and VV&A strategy. This entails performing the VV&A process on individual model modules composed of a specified version and configuration of a C4I application, operating system, and virtual machine. This will be referred to as a *C4I VM module* (Figure V-3). Alternatively, a module can be composed of the operating system and virtual machine sans the C4I application. This will be referred to as an *OS VM module* (Figure V-4). Regardless of the component make up, a modularized strategy (1) allows for flexibility when establishing C4I model networks, (2) encourages reuse of both the VM module and VV&A documentation, and (3) introduces efficiency in the VV&A process. Unfortunately, accrediting a reusable VM module implies an infrastructure to maximize its benefits and a dedicated accreditation team to update accreditations.

Deciding which VM module component structure to use (*C4I VM module* or *OS VM module*) presents their own set of advantages and disadvantages. C4I VM modules require more work for the VV&A team. Every new combination of each C4I application, OS, and VM version would have to go through the VV&A process. This method implies a dedicated team to accredit C4I VM modules. The issue is minimized, however, since documentation and processes will be reused. C4I VM modules also require access to developer test cases and scripts or the knowledge to create them. The advantage to this structure is that the end user is likely to have more confidence in the fidelity of the model since the C4I application was Validated and Verified while installed on the OS and VM.

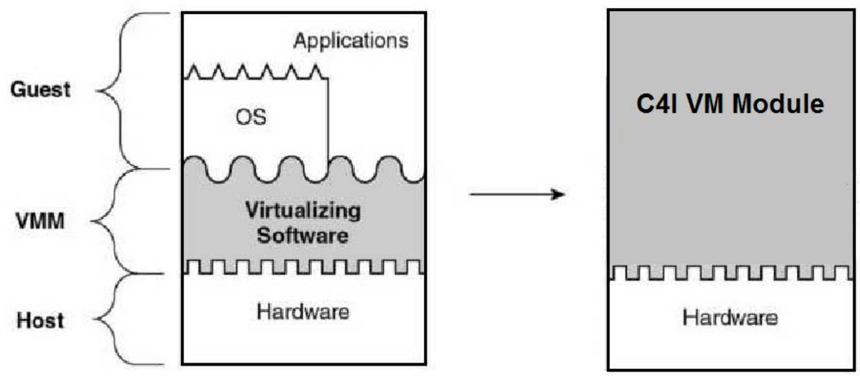

Figure V-4: C4I VM Module (After Smith & Nair, 2005)

Using the OS VM module, on the other hand, requires less time to accredit since established vendors already have OS and hardware compatibility databases that list out tested operating systems by the vendor's Quality Assurance. A VV&A practitioner could perform further benchmark or network analysis tests to add other layers of fidelity. This necessitates a less dedicated team and requires less VV&A iterations. The disadvantage to this structure is the level of confidence it provides an end-user. Depending on the M&S user's requirements and level of trust in virtualization technology, an OS VM module may or may not be enough to satisfy model credibility.

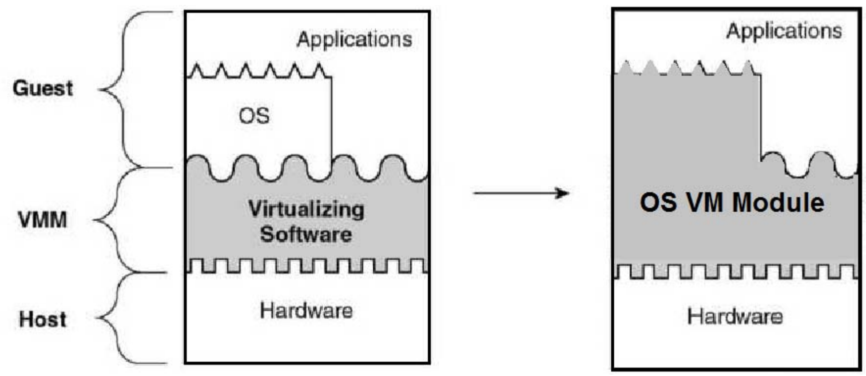

Figure V-5: OS VM Module (After Smith & Nair, 2005)

Irrespective of the VM module structure, the modularized strategy provides the much needed flexibility that DCSCs need since they often reconfigure C4I model networks to meet changing needs. For example, two C4I systems may be added to a classroom environment to accommodate additional students or an interoperability tester may need a network of two C4I systems one day and then require an Army Brigade's C4I architecture on another. Even environments such as MCTSSA's representative MEF C4I infrastructure are often reconfigured to meet the needs of individual tests. Combined with the already discussed advantages of virtual machines, an accredited VM module can be "mixed and matched" to meet the needs of individual tests despite such changing scenarios.

This strategy also exploits the advantages of reuse. This comes in two forms: component reuse and documentation reuse. Since virtual machines can be stored as software, an accredited VM module can be stored, duplicated, and reused by other DCSCs throughout the DoD. It matters little if the DCSC serves an acquisition community or a training community, the VM module can be reused with minimal overhead. Secondly, the VV&A documentation can be stored alongside the VM file as part of a whole package. DCSCs seeking to use the VM module can easily reference the documentation and (assuming that the documentation is sufficient) can reuse the work put into the VV&A process. This meets the policy dictated in DoD Directive 5000.59:

> M&S management shall develop plans, programs, procedures, issuances, and pursue common and cross-cutting M&S tools, data, and services to achieve DoD's goals by: promoting visibility and accessibility of models and simulations; leading, guiding, and shepherding investments in M&S; assisting collaborative research, development, acquisition, and operation of models and simulations; **maximizing commonality, reuse, interoperability, efficiencies and effectiveness of M&S, and supporting DoD Communities that are enabled by M&S.**

The flexibility and reusability ultimately leads to efficiencies in using VM models. Instead of having to accredit every permutation of a C4I network model, users can build their network composed of already accredited C4I modules. Testers, trainers, and analysts would not have to VV&A their ever changing networks themselves. Additionally, M&S users can focus less on building and configuring their environments since virtual machines come in prepackaged form. This frees them to focus instead on performing their analysis, training personnel, or testing C4I applications.

One disadvantage, however, is that the DoD requires an infrastructure, methodology, and awareness to maximize reuse amongst DCSCs. This limitation will not be addressed in this thesis and is recommended as future research in knowledge management. Another disadvantage is that a dedicated person or team would need to continually accredit

VM models components. The regular version updates of C4I applications or operating systems will require updated accreditations. Updated accreditation, however, is a necessity whenever a new C4I model environment is being established irrespective of modularization. Fortunately, the accreditation process for a version update is much faster than that of an entirely new C4I networked environment.

Modularizing the M&S and VV&A process is a sound strategy for those dealing with continually changing C4I network environments. Despite its disadvantages, the flexibility and reusability of a VM module (to include its VV&A documentation) saves time, effort, and resources overall. If a DCSC fails to modularize, it risks an unused VV&A process. M&S users often have limited resources and having to VV&A a C4I network configuration every time it changes will discourage future attempts to accredit their model. Accreditations will become outdated and meaningless. This, unfortunately, is the common state of VV&A in many DCSCs.

F. THE CASE FOR FUNCTIONAL TESTING

Since the purpose of the VV&A process is to assess and provide credibility for a model or simulation, one can argue that VV&A is not necessary for VMs if users are confident in the VM's ability to replicate a real computer. The use of VMs in production environments would support this argument. However, since many DCSCs are still new to the concept of virtualization, more rigorous V&V techniques need to be used to assuage critics. Referring back to Chapter II.E, the four main V&V method categories are Informal, Static, Dynamic, and Formal. Table II-1 further divides the multiple ways one can validate and verify a model or simulation. No one method is deemed the "correct" way for all V&V. Despite the lack of a standard operating practice, the method must answer the question: Does it support the credibility of the M&S? The VV&A Recommended Practices Guide (Dobey et al., 2006) states that credibility is determined by a model's "capabilities and correctness, the accuracy of its results, and its usability in the specified application".

This thesis recommends functional testing for the V&V method of C4I VM modules. It is a commercially accepted test method for industry level software applications and is a common method for Quality Assurance professionals. This recommendation is due to four main criteria: (1) M&S users require rapid access to their models, (2) DoD V&V professionals are ill equipped and ill numbered to perform more detailed testing (e.g., white-box testing), (3) V&V practitioners can reuse a C4I application developer's test cases/methods, and (4) C4I environment models tend to be focused on interoperability and integration. The VV&A Recommended Practices Guide describes functional testing as the following:

> Functional testing (also called ***black-box testing***) assesses the accuracy of model input-output transformation. It is applied by inputting test data to the model and evaluating the accuracy of the corresponding outputs. It is virtually impossible to test all input-output transformation paths for a reasonably large and complex simulation because the paths could number in the millions. Therefore, the objective of functional testing is to increase confidence in

> model input-output transformation accuracy as much as possible rather than to claim absolute correctness. (Dobey et al., 2006)

Because black-box testing is less comprehensive than white-box testing (i.e., a method that tests an application's inner-workings and structures such as path testing or data flow testing), it consumes less time thus allowing users faster access to needed models. This method also leverages the vendor's (whether it be VM, operating system, or C4I application) test and Q&A processes without going on the extreme of fully trusting the software. Choosing a more time-consuming test method risks the avoidance of the VV&A process as M&S users push to meet deadline requirements.

Secondly, DoD V&V professionals are often ill equipped and ill numbered to perform more detailed testing (e.g., white-box testing). Methods such as white-box testing require specialized skills such computer programming which are often reserved for actual development work. Even with the proper skill set, vendors will very rarely allow outsiders to examine their software's inner designs. Finally, the manpower requirements to perform more detailed testing will far exceed VV&A personnel numbers.

Another support for performing functional testing on a C4I module is that V&V practitioners can reuse a C4I application developer's test cases/methods. Assuming that such transactions are permissible in contract (or are negotiable) and that the C4I developer performed repeatable tests, the V&V practitioner can run tests on a virtualized C4I component and compare them against physical C4I component (i.e., a C4I application installed on a physical computer). This saves time and effort by leveraging the advantages of reuse.

Finally, black-box testing is well suited for DCSCs because the organizations that need the advantages of virtualized environments tend to be focused on interoperability and integration rather than hardware compatibility testing. Because of this, M&S users are more interested in the fact that a C4I application will accurately send information to another application in the network given a set of inputs. As quoted earlier, "the objective of functional testing is to increase confidence in model input-output transformation accuracy as much as possible rather than to claim absolute correctness" (Dobey et al., 2006). For example, C4I training facilities teach application usage and how the applications work together.

The largest disadvantage however is that black-box testing of most modern C4I applications have too many transformation paths and it is virtually impossible to test them all:

> Generating test data is a crucially important but very difficult task. The law of large numbers does not apply. Successfully testing the model under 1,000 input values (i.e., test data) does not imply high confidence in model input-output transformation accuracy just because the number appears large. Instead, the number of input values used should be compared with the number of allowable input values to determine the percentage of the model input domain that is covered in testing. The more the model input domain is covered in testing, the more confidence is gained in the accuracy of the model input-output transformation (Dobey et al., 2006)

When compared to white-box testing (the other common software testing method) however, black-box testing is more feasible. The complexity of modern applications will not allow a fully comprehensive testing of every line of code and logic path as required in white-box testing.

Despite the limitation of black-box testing, it is a commercially accepted test method to test the functional capability of production software. When one puts into context that (1) M&S users need rapid access to their models, (2) DoD V&V professionals are ill equipped and ill numbered to perform more detailed testing, (3) V&V practitioners can reuse C4I developer test cases/methods, and (4) C4I environment models tend to be focused on interoperability and integration, it becomes apparent that functional testing is an ideal V&V method to provide credibility to a C4I model. A decision to use a less stringent method risks a lack of use because users will not have enough confidence in the credibility of the model. A more stringent method, on the other hand, risks a burdensome VV&A process that cannot cope with the ever-changing Information Technology market, and will unnecessarily delay the deployment of needed technology to units.

G. THE RECOMMENDED PROCESS

Given the arguments listed in this chapter, this thesis proposes a process to help tailor the VV&A process of a virtualized C4I environment while taking advantage of reusability for future accreditation needs. Since DCSCs differ in structure, skill set, and requirements, the following process is a guideline to be tailored to the organization. Before going through the process, it is important to perform a risk assessment on the COTS VM to be used when going through the acquisition process. This helps ensure that the VM product will have the needed support, personnel, training, and maturity to meet DCSC needs --- assumptions that are made for the process to succeed. Another assumption made is that VM module users use the same COTS VM technology (i.e., same vendor).

The needs and capabilities of a DCSC will dictate whether a C4I VM module or an OS VM module will be used. Acquisition or test DCSCs are more likely to have the skill set, accessibility to developer test cases/scripts, and personnel to build C4I VM modules. OS VM modules can and should also be built for those that find the COTS VM as "credible" since it provides the most flexibility for end-users. Once a VM product is chosen, acquired, and established in the DCSC, the following process should be followed:

Step 1: Build the baseline. This computer will be used for comparative purposes. If performing VV&A on an OS VM module, go to Step 1a. If performing VV&A on a C4I VM module, go to Step 1b.

Step 1a: Install OS on physical computer. Follow OS installation guide published by OS vendor.

Step 1b: Install C4I application on physical computer. Follow OS installation guide published by OS vendor. Then follow the installation guide published by C4I vendor on the OS.

Step 2: Build the VM module. This is the actual VM module that will undergo the VV&A process. If performing VV&A on an OS VM module, go to Step 2a. If performing VV&A on a C4I VM module, go to Step 2b.

Step 2a: Build OS VM module. Set VM configurations to match the modeled hardware (i.e., RAM, hard drive capacity, NIC speed, CPU speed, etc.) in Step 1a. Follow OS installation guide published by COTS VM vendor and OS vendor. OS installation steps should match those of Step 1a.

Step 2b: Build C4I VM module. Set VM configurations to match the modeled hardware (i.e., RAM, hard drive capacity, NIC speed, CPU speed, etc.) in Step 1b. Follow OS installation guide published by COTS VM vendor and OS vendor. Then follow C4I application installation guide published by C4I vendor on the OS.

Step 3: Perform VV&A on the VM module. Use the DVDT (DoD VV&A Documentation Tool) for documentation purposes. This tool hastens and standardizes the documentation process in addition to aiding documentation reusability. Standard VV&A process is followed using Functional Testing as the V&V method. If performing VV&A on an OS VM module, go to Step 3a. If performing VV&A on a C4I VM module, go to Step 3b.

Step 3a: Ensure that the VM vendor supports the operating system. Custom scripts can be used to compare correctness for added credibility. In addition, one can use CPU benchmarking tools (such as BapCO SYSmark or SPEC CPU), network benchmarking tools (such as Netperf or NetSpec), or Input-output benchmarks (such as IOZone) for added credibility. Be mindful, however, that benchmarks tend to compare a VM's performance against a real computer --- not its correctness. Run tool/s on the VM and the physical computer concurrently for results comparison. Compare outputs and validate results.

Step 3b: Retrieve test cases/scripts from C4I developer. If none available, develop test cases/scripts internally. Use stubs to capture outputs (e.g., network sniffers for network outputs). One can use CPU benchmarking tools (such as BapCO SYSmark or SPEC CPU), network benchmarking tools (such as Netperf or NetSpec), or Input-output benchmarks (such as IOZone) for added credibility. Be mindful, however, that benchmarks compare a VM's performance against a real computer --- not its correctness. Run tests on C4I VM component model and physical computer concurrently for results comparison.

Step 4: Configuration Control and publish. Once the model is accredited, turn in the VM module and VV&A documentation to Configuration Control. Publish its availability to interested personnel or community of interest.

Step 5: Use VM module. M&S Users download component. Verify that component meets M&S needs via VV&A documentation. Use components as needed.

Step 1 to 4 is a looped process. Whenever a new OS version or a new C4I application version (for C4I VM modules), steps 1 to 4 are repeated. Alternatively, the process can be performed on an as-needed basis. This would reduce the manpower overhead of accrediting every version, but it also reduces the lead-time before an M&S user can use a VM module for use if one does not already exist. Subsequent iterations will reuse documentation and test cases/scripts. VV&A practitioners make modifications as necessary.

VI. USE CASE: MCTSSA

This chapter will use Marine Corps Tactical Systems Support Activity (MCTSSA) as a sample DCSC in order to exemplify concepts discussed in the previous chapter. It will first introduce MCTSSA, its mission, and its structure. This chapter will then discuss its trend towards virtualization and assess its chosen virtualization technology. Next, the chapter will go through the steps of developing a C4I VM module and go through the VV&A process. This chapter will not aim to go through the entire process in detail but instead augment the previous chapters with details specific to MCTSSA.

A. ABOUT MCTSSA

Marine Corps Tactical Systems Support Activity is a DCSC that acts as the MAGTF (Marine Air Ground Task Force) C4I Systems Engineering Interoperability, Architecture, and Technology (SIAT) center. According to its website (MCTSSA, 2010), its mission is to validate and verify MAGTF systems integration and interoperability. It is a component of Marine Corps Systems Command (MARCORSYSCOM) and supports three primary customers:

- Commanding General, MARCORSYCOM, and Program Managers to acquire and sustain C4ISR products for the Operating Forces.
- Operating Forces for fielded C4 systems.
- Deputy Commander SIAT, MARCORSYSCOM, for C4ISR systems engineering and integration.

MCTSSA is an ideal use case because it has multiple sub-entities that provide C4I support. It also has the skill set to perform functional testing on the VM modules.

- <u>Operating Forces Tactical Systems Support Center (OFTSC)</u>. It serves as the single point of entry for resolving C4I systems problems for the operating forces. (MCTSSA, 2010)
- <u>Program and Engineering Support Group (PESG)</u>. It provides technical support to the Commander, MARCORSYSCOM, and Program Managers to acquire and sustain C4I Systems for the Operating Forces. (MCTSSA, 2010)
- <u>Test and Certification Group (T&CG)</u>. It provides technical support to the Commander, MARCORSYSCOM, and Program Managers for Testing and Certification of C4 Tactical Systems. (MCTSSA, 2010)
- <u>Technical Infrastructure and Support Group (TI&SG)</u>. It provides USMC decision makers with interoperability and integration assessments of Command, Control, Computer, Communications, Intelligence, Reconnaissance, and Surveillance (C4ISR) systems. (MCTSSA, 2010)

B. THE TREND TOWARDS VIRTUALIZATION

In 2007, the Marine Corps signed a contract with VMware (Ferguson, 2007). Since then, units and organizations have begun to implement the technology in varying degrees of scope and effectiveness. MCTSSA, also, has small isolated pilots of virtualized environments. For example, OFTSC currently employs a VM sandbox lab and T&CG has a virtual environment for validating and vetting test threads (Capt Mayo, 2009). Despite these advances, there are still reservations with employing virtual machines thus making MCTSSA an appropriate use case for using VV&A to accredit VM modules. MCTSSA also makes for an interesting use case because an accredited VM module can be shared amongst its divisions to maximize reusability.

Despite the current contract between the Marine Corps and VMware, it is a good practice to perform a COTS risk assessment on VMware's virtual machine technology. It will reveal whether further investment is beneficial before continuing (Appendix C). Confirming that the COTS product is a low-risk acquisition, next steps involve creation of the C4I VM module and going through the VV&A process.

C. DEVELOP A C4I VM MODULE

Since virtualization is still an "unproven" technology to MCTSSA, accrediting C4I VM modules (as opposed to OS VM modules) is the recommended method. Confidence must be earned that C4I applications will run the same way in the virtual environment as it will in the physical. Their tests require input-output fidelity as well as performance fidelity.

D. VV&A ROLES

The following are potential delegations of responsibility for VV&A roles. Role descriptions can be referenced in paragraph II.F.

- **M&S User** – Test Engineers and OFTSSC helpdesk personnel from MCTSSA are the primary users of the C4I VM module for their respective missions.
- **Accreditation Authority** – The MCTSSA Commanding Officer is the final approval authority for the accreditation of the C4I VM module.
- **Accreditation Agent** – The T&CG group at MCTSSA would perform the accreditation assessment for the M&S. They have the knowledge on C4I application expected behavior and would be able to provide the best recommendation.
- **M&S Proponent** – The Information System & Infrastructure Program Office (PG-10) would be the proponent for the use of C4I VM module and virtualization environments. This role can be delegated to the PESG group aboard MCTSSA since they support the Program Office.
- **V&V Agent** – This can be performed by contractors or personnel from the T&CG group at MCTSSA.
- **M&S Developer** – Since a COTS VM solution is being used, VMware is the primary M&S Developer. Personnel from the TI&SG group would establish the actual virtualization environment (i.e., C4I VM module creation, networking, etc.).
- **Subject Matter Expert** – VM experts, C4I application experts, and test procedure experts would play assist roles for this process.

E. UNDERGOING THE PROCESS

In addition to undergoing the VV&A process (paragraph II.D), using the DVDT, applying the functional test V&V method (paragraph V.F), and recommended C4I VM module process (paragraph V.G), one of the specific needs of MCTSSA is the ability to undergo performance testing. To compensate for this, the following additional tests should be performed:

- Use CPU benchmarking software to compare real and virtual system performance. Example: Performance Test (http://www.passmark.com/)

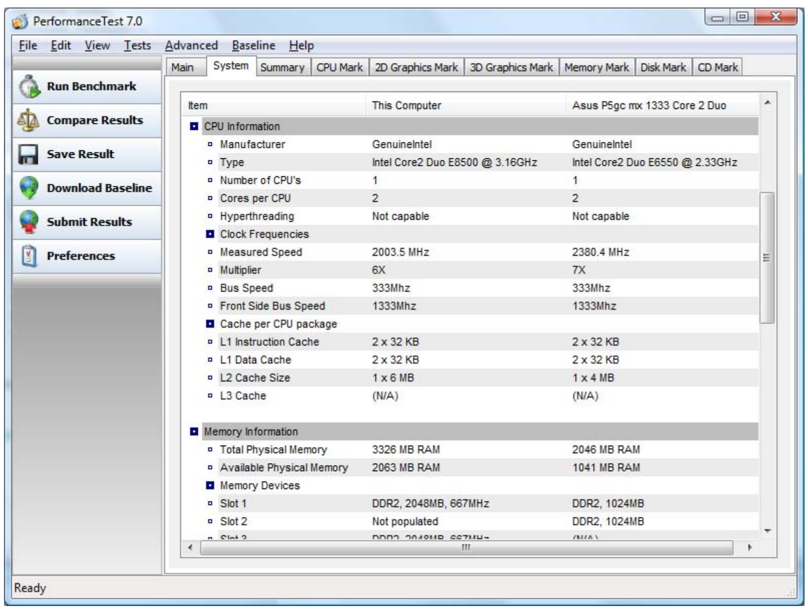

Figure VI-1: PerformanceTest 7.0 Screenshot

- Use a network limiter to simulate slow networks such as WANS or congestion. Example: NetLimiter (http://netlimiter.com/)

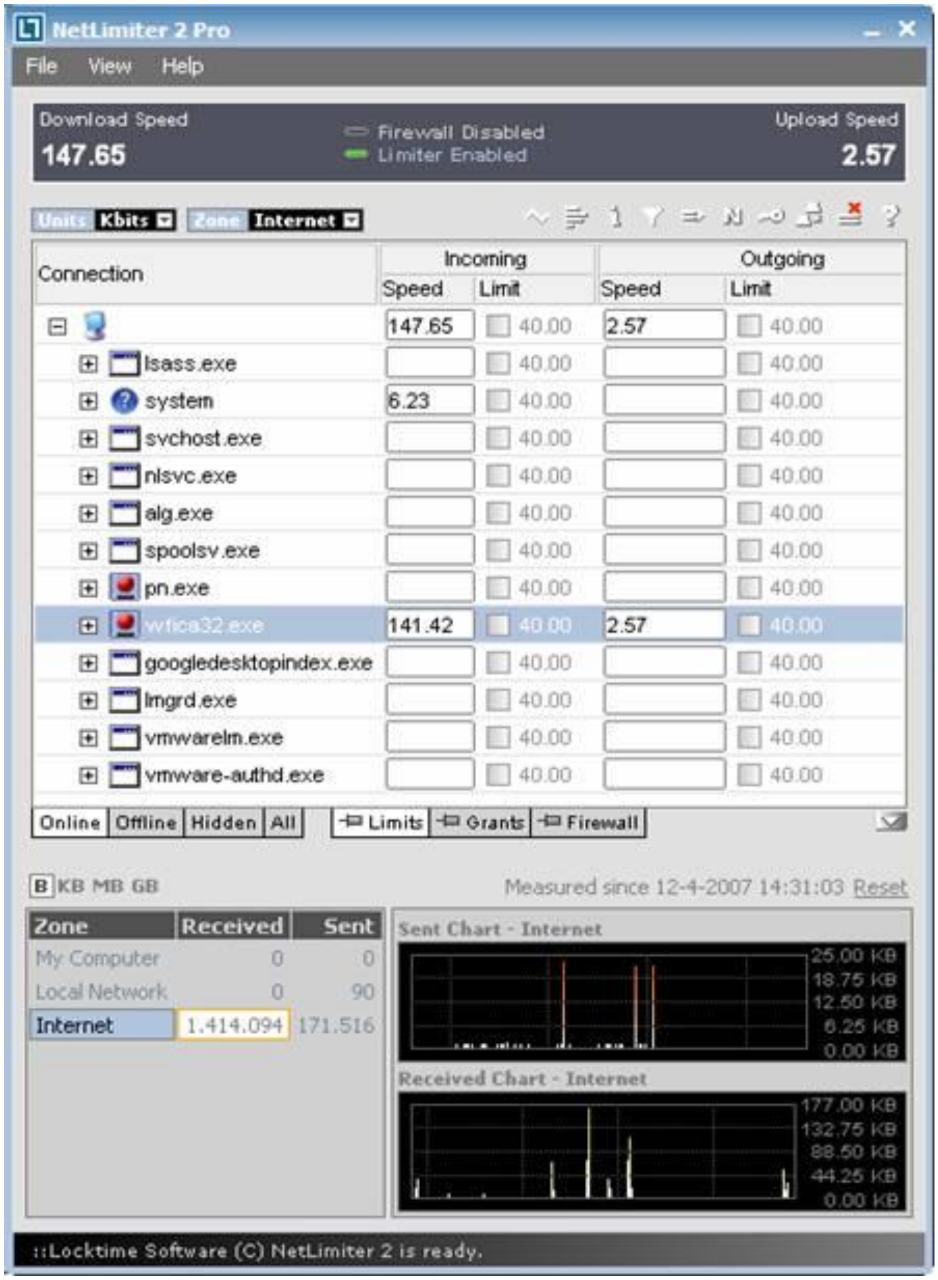

Figure VI-3: NetLimiter screenshot

- Use a network simulator to reproduce latency introduced by WANs or satellite connections. Example: TMnetsim (http://tmurgent.com/Tools.aspx)

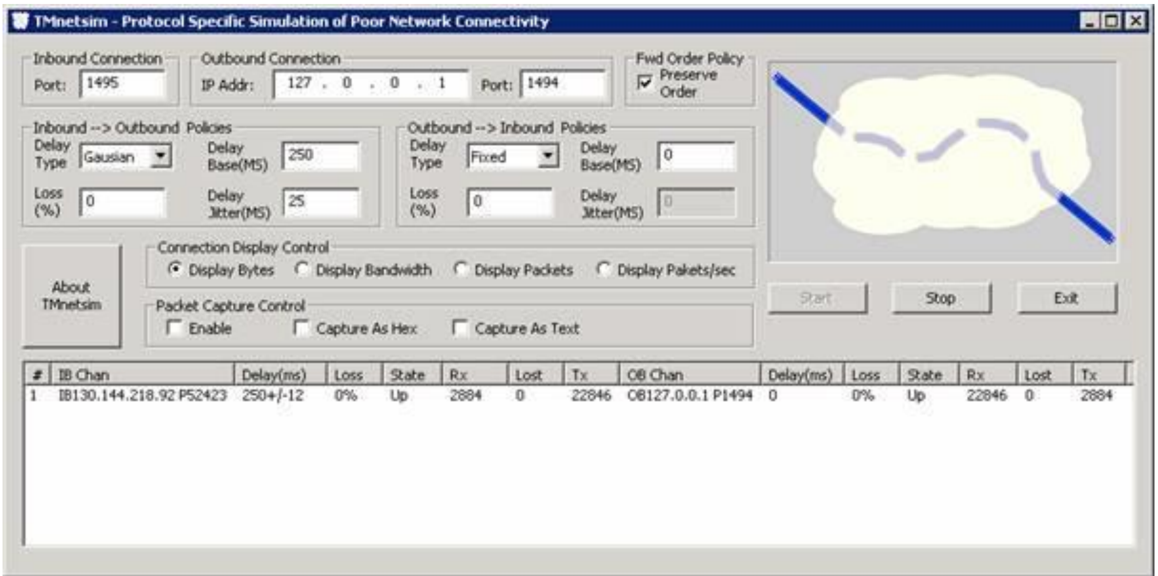

Figure VI-2: TMnetsim screenshot

F. C4I VM MODULE REUSE

Since MCTSSA has multiple C4I support functions, it can take immediate advantage of C4I VM module reuse. Once accredited, it can be used by T&CG for future tests and by OFTSCC when supporting deployed units with C4I problems. Outside of MCTSSA, the C4I VM module can be given to other M&S communities for use (e.g., Operations Analysis Division, MCCDC; Marine Corps Warfighting Laboratory, MCCDC; Marine Corps Operational Test & Evaluation Activity, Training and Education Command). Although outside the scope of this thesis, the C4I VM module can conceivably be used by deployed forces (paragraph VII.B - Recommendations for Future Research).

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION

A. SUMMARY

Due to the growing demand for and the increasing number of C4I systems, DCSCs are forced to maintain rooms or facilities that model C4I networks. A training command may only need a classroom with networked C4I systems (i.e., as a training model) to train Operations staff while an interoperability test center may have a network of disparate DoD organizations creating joint C4I network model (i.e., as a testing model). If these centers are fortunate, Program Offices carry acquisition costs of the actual test or training systems as part of the budget. Unfortunately, DCSCs still have to shoulder the costs of maintenance, energy, cooling, manpower, infrastructure, and storage costs inherent to maintaining C4I test environments. These encompass the greater bulk of total costs of ownership (Figure II-1). DCSCs often have very limited budgets and cannot effectively meet their missions with such growing total costs of ownership. Its current methods for modeling C4I systems need to be rethought if they aim to meet their goals.

Irrespective of a DCSC’s specific needs, VMs are generally ideal C4I environments. They provide repeatable, controlled, safe, and inexpensive environments. When factoring in the reality that DCSCs are support organizations and are often less prioritized than deployable combat units, the allure of cost-savings coupled with greater capability is strong. Despite the limitation that a

VM does not have the fidelity of a real computer, organizations have come to realize the compelling need to use virtual machines. Software developers and enterprise-level companies routinely use virtual machines to develop and test applications before using them in production environments. Many are even using virtual machines as production environments.

The use of a commercial-off-the-shelf solution for virtual environments is an obvious answer to those considering the technology. The marketplace is healthy with competition and the technology is mature. The COTS implementation also allows DCSCs to outsource the M&S process as well as the M&S role in the VV&A process. Leveraging the vendor's expertise and production capabilities in this arena allows DCSCs to focus on the important technologies and implementations specific to their needs. Although manpower overhead and the inability to virtualize non-standard C4I hardware are disadvantages, they are not overwhelming hurdles and the advantages of COTS virtualization easily make up for any shortfalls.

Classic system VMs are the most logical virtualization technology for use by DCSCs. It has the right balance of fidelity, portability, and repeatability needed. DCSCs can take advantage of the economies of scale by standardizing on a single technology and can leverage the benefits of a mature technology market. Choosing a different virtualization technology such as process VMs or whole-system VMs risks failing fidelity requirements needed by certain DCSCs. If this occurs, DCSCs will not be able to exploit the advantages of shared technology such as economies of scale.

Modularizing the M&S and VV&A process is a sound strategy for those dealing with continually changing C4I network environments. Despite its disadvantages, the flexibility and reusability of a VM module (to include its VV&A documentation) saves time, effort, and resources overall. If a DCSC fails to modularize, it risks an unused VV&A process. M&S users often have limited resources and having to VV&A a C4I network configuration every time it changes will discourage future attempts to accredit their model. Accreditations will then become outdated and meaningless.

Functional testing ("black-box testing") is also an ideal V&V method for providing credibility for VM modules. When one puts into context that (1) M&S users need rapid access to their models, (2) DoD V&V professionals are ill equipped and ill numbered to perform more detailed testing, (3) V&V practitioners can reuse C4I developer test cases/methods, and (4) C4I environment models tend to be focused on interoperability and integration, it becomes apparent that functional testing is an ideal V&V method to provide credibility to a C4I model. A decision to use a less stringent method risks a lack of use because users will not have enough confidence in the credibility of the model. A more stringent method, on the other hand, risks a burdensome VV&A process that cannot cope with the ever changing Information Technology market.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

Based on the numerous benefits on the reusability of VM module, research should be performed in establishing an infrastructure to increase awareness and actual reuse by various DCSCs throughout the DoD. In addition, the C4I VM component concept can be expanded throughout the entire product cycle of a C4I application. Research should be performed on the use of C4I VM modules from inception to production.

APPENDIX A

Hyper-V Server 2008 R2 from (From Microsoft, 2010)

	Virtualization Needs	Microsoft Hyper-V Server 2008 R2	Windows Server 2008 R2 Standard	Windows Server 2008 R2 Enterprise	Windows Server 2008 R2 Datacenter
Scenarios	Server Consolidation	✔	✔	✔	✔
	Test and Development	✔	✔	✔	✔
	Branch Server Consolidation	✔	✔	✔	✔
	Virtual Desktop Infrastructure (VDI)	✔		✔	✔
	Mixed OS virtualization (Linux and Windows)	✔	✔	✔	✔
	Dynamic Data Center			✔	✔
Features	Host Clustering	✔		✔	✔
	Live Migration	✔		✔	✔
	Large Memory support (Host OS) > 32GB	✔		✔	✔
	Support for >4 Processors (Host OS)	✔		✔	✔
	Local Graphical User Interface		✔	✔	✔
	Ability to Add Additional Server Roles		✔	✔	✔
	Guest Virtualization Rights Included in Host Server License		✔	✔	✔
	Application Failover			✔	✔

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B

VSphere Editions from (From VMware, 2010a)

	Standard	Advanced	Enterprise	Enterprise Plus
Product Components				
Memory/Physical Server	256GB	256GB	256GB	No Memory Limit
Cores per Processor	6	12	6	12
Processor Support	Per 1 CPU	Per 1 CPU	Per 1 CPU	Per 1 CPU
Centralized Management Compatibility				
vCenter Compatibility (Sold Separately)	-vCenter Foundation -vCenter Standard	-vCenter Foundation -vCenter Standard	-vCenter Foundation -vCenter Standard	-vCenter Foundation -vCenter Standard
Product Features				
Thin Provisioning	✓	✓	✓	✓
Update Manager	✓	✓	✓	✓
Data Recovery	Sold Separately for this Edition	✓	✓	✓
High Availability	✓	✓	✓	✓
vMotion	✓	✓	✓	✓
vStorage APIs for Data Protection	✓	✓	✓	✓
Virtual Serial Port Concentrator		✓	✓	✓
Hot Add		✓	✓	✓
vShield Zones		✓	✓	✓
Fault Tolerance		✓	✓	✓
vStorage APIs for Array Integration			✓	✓
vStorage APIs for Multipathing			✓	✓
Storage vMotion			✓	✓
Distributed Resources Scheduler (DRS), Distributed Power Management (DPM)			✓	✓
Storage I/O Control				✓
Network I/O Control				✓
Distributed Switch				✓
Host Profiles				✓

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C

Risk Assessment Chart (After Hensley, 2000)

RISK ASSESSMENT CHART					
Product Name/Version: VMware VSPhere 4.0			**Assessment Date:**		Rating
			Assessed By:		
Risk Category	**Risk Factor**	**Risk Cues**			
		Low	**Medium**	**High**	
Technology	Maturity/Stability	Widely accepted technology.	Competing technologies.	Emerging technology.	L
	Competition	Large number of competing products within the selected technology.	Limited number of competing products within the selected technology.	Small number of competing products or no competition within the selected technology.	L
Vendor	Maturity/Stability	Large company. Applies commercially accepted development practices.	Medium company. Applies a mix of commercially accepted and ad-hoc development practices.	Small/emerging company. Applies ad-hoc development practices.	L
	Technology Expertise	Maintains personnel base with expertise in the technology.	Access to personnel with technology expertise. Moving into an emerging technology.	Limited or no access to personnel with technology expertise.	L
	Responsiveness	Accepts/processes customer feedback. Provides advance notice of product changes.	Accepts/processes market feedback. Provides limited notice of product changes.	Does not accept/process customer feedback. Provides no notice of product changes.	L
	Technical Support	Maintains knowledgeable technical support staff. Maintains 24/7 help desk. Easy access to help desk. Easy access to patches.	Maintains semi-knowledgeable technical support staff. Restricted help desk availability. Limited avenues to access help desk. Limited access to patches.	Knowledgeable technical assistance staff not available. No help desk. No access to patches.	L
Product	Market Acceptance	Wide market acceptance. Large market share. Product drives the market.	Limited market acceptance. Medium market share.	Product not widely accepted by the market. Small market share.	L
	Stability/Robustness	Very few significant upgrades. No significant bugs or limited insignificant bugs.	Moderate number of product upgrades/patches. Tolerable bugs (non-critical).	Significant number of product upgrades/patches. Significant or intolerable bugs.	L
	Interfaces	Uses commercially accepted interfaces. Interface documentation is available.	Uses a mix of commercially accepted interfaces and nonstandard or proprietary interfaces. Limited interface documentation.	Uses nonstandard or proprietary interfaces. No interface documentation.	L
	Complexity/Features	Easy to use. Easy to install and configure. Few extraneous capabilities. No undesirable features.	Moderately easy to use. Moderately easy to install or configure. Some extraneous capabilities. May have an undesirable feature.	Hard to use. Difficult to install or configure. Large number of extraneous capabilities. Exhibits undesirable features.	L
	Security	No significant security issues. No insignificant security issues.	No significant security issues. A few insignificant security issues.	Significant security issues. Many insignificant security issues.	L
	Safety	No safety issues.	N/A	Safety issue.	L
	Documentation	Understandable, complete, and accurate documentation package.	Acceptable documentation package. Falls short in some areas.	Poor documentation package.	L
	Cost	Competitive product cost. Good warranty. Reasonable maintenance fees.	Inflated product cost. Poor warranty. Inflated maintenance fees.	Unreasonable product cost. No warranty. Unreasonable maintenance fees.	L

NOTES:

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Belady, C. L. (2007). In the data center, power and cooling costs more than the IT equipment it supports. Retrieved August 10, 2010, from http://www.electronics-cooling.com/2007/02/in-the-data-center-power-and-cooling-costs-more-than-the-it-equipment-it-supports/

Mayo, D. (2009). *MCTSSA tactical and enterprise virtualization*. MCTSSA, Camp Pendleton, CA.

Connor, D. (May 15, 2008). Power costs drive moves to virtual servers. Retrieved August 23, 2010, from http://www.cio.com/article/358863/Power_Costs_Drive_Moves_to_Virtual_Servers

Dobey, V., Cross, D., Harmon, S., Lewis, R., Metz, M., Pace, D., Senko, R., & Solick, S. (2006). VV&A recommended practices guide. Retrieved July 25, 2010, from http://vva.msco.mil

DoD Directive 5000.1. (2003). The *defense acquisition system*. Washington, DC: U.S. Government Printing Office.

DoD Directive 5000.59. (2007). *DoD modeling and simulation (M&S) management*. Washington, DC: U.S. Government Printing Office.

DoD Directive 5000.61. (2009). *DoD modeling and simulation (M&S) verification, validation, and accreditation (VV&A)*. Washington, DC: U.S. Government Printing Office.

Ferguson, S. (2007). Marines bring virtualization to the battlefield. Retrieved August 22, 2010, from http://www.eweek.com/c/a/IT-Infrastructure/Marines-Bring-Virtualization-to-the-Battlefield/

Galorath, D. D., & Evans, M. W. (2006). *Software sizing, estimation, and risk management*. Boca Raton, New York: Auerbach Publications, Hensley, B. J. (2000). *Development of a software evolution process for military systems composed of integrated commercial off the shelf components.* Master's thesis, Naval Postgraduate School.

Mann, J. (2007). Virtualization market to hit $11.7 billion by 2011. Retrieved August 8, 2010, from http://www.techspot.com/news/26103-virtualization-market-to-hit-117-billion-by-2011.html

McKinney, D. (2001). Impact of commercial-off-the-shelf (COTS) software and technology on systems engineering. Presentation to INCOSE Chapters,

MCTSSA (2010). Marine Corps Tactical Systems Support Activity. Retrieved February 11, 2010, from http://www.mctssa.usmc.mil/

Microsoft (2010). Hyper-V server 2008 R2. Retrieved August 22, 2010, from http://www.microsoft.com/hyper-v-server/en/us/default.aspx

MSIAC (2010). A modeling and simulation primer. Retrieved Aug 7, 2010, from http://education.dod-msiac.org/ms_primer.asp?a=s4&b=view&c1=272

Navy Modeling and Simulation Management Office. (2004). Modeling and simulation verification, validation, and accreditation implementation handbook, Vol 1 Department of Navy. Washington, DC: U.S. Government Printing Office.

Office of the Secretary of Defense. (2000). Commercial item acquisition: considerations and lessons learned. Washington, DC: U.S. Government Printing Office.

Sanders, P., & Office of the Undersecretary of Defense (A&T) Washington DC. (1996). DoD modeling and simulation (M&S) verification, validation, and accreditation (VV&A). Washington, DC: U.S. Government Printing Office.

Smith, J. E., & Nair, R. (2005). Virtual machines: versatile platforms for systems and processes (1st ed.). San Francisco, California: Morgan Kaufmann.

VMware. (2010a). Compare vSphere editions for mid-size and enterprise businesses. Retrieved August 22, 2010, from http://www.vmware.com/products/vsphere/buy/editions_comparison.html

VMware. (2010b). VMware customers. Retrieved February 11, 2010, from http://www.vmware.com/company/customers/

Webster, B. F. (2008). Buy vs. build software applications: The eternal dilemma. Retrieved July 25, 2010, from http://www.baselinemag.com/c/a/Application-Development/Buy-vs-Build-Software-Applications-The-Eternal-Dilemma/

Yegulalp, S. (2010). Hypervisor. Retrieved August 5, 2010, from http://itmanagement.earthweb.com/entdev/article.php/3880086/Hypervisor.htm

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education,
   MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center,
   MCCDC Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity
   Attn: Operations Officer
   Camp Pendleton, California

7. Chairman, Department of Information Sciences,
   Naval Postgraduate School
   Monterey, California